# Validation of Monte Carlo Estimation of Problem Discovery Likelihood

TR 29.3357

James R. Lewis

Speech Product Design and Usability

West Palm Beach, Florida

## Abstract

This experiment compared estimation of problem discovery likelihood (*p*) using both factorial combination and Monte Carlo sampling. Factorial combination assures accurate estimation, but faces serious computational limits when run on standard personal computers. Monte Carlo sampling is less intensive with regard to memory limitations, but does not necessarily assure accurate estimation because it relies on random sampling of combinations. I used both methods to estimate *p* from patterns of problems reported in several published usability evaluations. The results of the experiment showed that Monte Carlo sampling with 1000 iterations produced essentially the same results as complete factorial combination.

## ITIRC Keywords

Monte Carlo estimation
factorial combination
problem discovery likelihood
usability evaluation
sample size estimation

## Contents

# Introduction

## The Problem

Investigations into sample size estimation have found the *p*, the likelihood of problem discovery for a product or system undergoing usability evaluation, plays a key role in determining the required sample size for a usability study (Lewis, 1994). Following the practice of using pilot studies to estimate variability when planning sample sizes for experiments based on comparison of means (Diamond, 1981; Walpole, 1976), some authors have recommended getting estimates of *p* from small sample usability studies for the purpose of estimating usability study sample sizes (Lewis, 1991, 2000). Recently, though, Hertzum and Jacobsen (in press) have pointed out that this practice will almost always result in overestimation of the value of *p*.

For example, consider the distribution of discovered problems across participants in Table 1. An 'x' in the table indicates that this participant experienced this problem during the usability evaluation. In this hypothetical example, all participants experienced Problem 1, but only the first and tenth participants experienced Problem 10. Because the entire matrix has 100 cells (ten participants by ten problems) and 50 cells contain an 'x', the value of *p* is .5 (50/100). Note that this is the same as the estimate of *p* calculated by averaging *p* for each participant in the table.

*Table 1. Hypothetical distribution of ten usability problems over ten participants*

| ***Participant*** | **Prob 1** | **Prob 2** | **Prob 3** | **Prob 4** | **Prob 5** | **Prob 6** | **Prob 7** | **Prob 8** | **Prob 9** | **Prob 10** | **Count** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | x | x | | x | | x | | x | | x | 6 |
| *2* | x | x | | x | | x | | x | | | 5 |
| *3* | x | x | | x | x | x | | | | | 5 |
| *4* | x | x | | x | | | x | | | | 4 |
| *5* | x | x | x | x | | x | | | x | | 6 |
| *6* | x | x | x | | | | | x | | | 4 |
| *7* | x | x | x | | x | | | | | | 4 |
| *8* | x | x | x | | x | | x | | | | 5 |
| *9* | x | | x | | x | | x | | x | | 5 |
| *10* | x | | x | | x | | x | | x | x | 6 |
| | | | | | | | | | | | |
| *Count* | 10 | 8 | 6 | 5 | 5 | 4 | 4 | 3 | 3 | 2 | 50 |

Suppose, though, that in this hypothetical example the usability practitioner had stopped the evaluation after the third participant. In that case, the known distribution of problems would be a subset of the set of problems discovered with ten participants, as shown in Table 2.

*Table 2. Hypothetical distribution of problems discovered with first three participants*

| *Participant* | Prob 1 | Prob 2 | Prob 3 | Prob 4 | Prob 5 | Prob 6 | Prob 7 | Prob 8 | Prob 9 | Prob 10 | Count |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | x | x | | x | | x | | x | | x | 6 |
| *2* | x | x | | x | | x | | x | | | 5 |
| *3* | x | x | | x | x | x | | | | | 5 |
| | | | | | | | | | | | |
| *Count* | 3 | 3 | 0 | 3 | 1 | 3 | 0 | 2 | 0 | 1 | 16 |

In Table 2, there are 30 cells (three participants by ten problems) and 16 cells containing 'x'. Dividing the number of cells containing 'x' by the total number of cells produces .533 as the estimate of *p* (which isn't much different from the estimate derived from Table 1). In this case, however, the practitioner would not know of the existence of Problems 3, 7, and 9 because none of the first three participants experienced these problems. So, when the practitioner would gather the data together for the purpose of estimating *p*, the data would not contain those columns, as shown in Table 3.

*Table 3. Hypothetical problem distribution with three participants: practitioner's view*

| *Participant* | Prob 1 | Prob 2 | Prob 4 | Prob 5 | Prob 6 | Prob 8 | Prob 10 | Count |
|---|---|---|---|---|---|---|---|---|
| *1* | x | x | x | | x | x | x | 6 |
| *2* | x | x | x | | x | x | | 5 |
| *3* | x | x | x | x | x | | | 5 |
| | | | | | | | | |
| *Count* | 3 | 3 | 3 | 1 | 3 | 2 | 1 | 16 |

In Table 3, there are only 21 cells (seven observed problems by three participants), with sixteen of the cells containing an 'x'. This reduction in the denominator increases the estimate of *p* from .533 to .762, about a 50% overestimation.

This is a potentially serious problem because overestimation of *p* necessarily leads to underestimation of the required sample size. The consequence of undersampling would be to fail to achieve the problem discovery goals for a usability study.

## Approaches to Investigating the Problem

Fortunately, over the last ten years a number of researchers have published the distribution of problems discovered in usability evaluations with fairly large samples (Lewis, 1994; Nielsen & Molich, 1990; Virzi, 1990). These distributions provide a source for conducting investigations of the overestimation of *p* as a function of pilot sample size and the true value of *p*. There are two approaches to take when using these distributions for this type of investigation: factorial combination and Monte Carlo sampling.

To apply the factorial combination approach, it is necessary to compute *p* for every possible combination of participants from a problem discovery distribution. The formula for computing the total number of combinations for *n* items taken *r* at a time is $C = n!/(r!(n-r)!)$ (Bradley, 1976). If a study contains ten participants and an investigator wants to estimate the value of *p* for every combination of two participants, then it is necessary to compute *p* for the 45 different ways of selecting two objects from a set of 10 (10!/(2!8!) reduces to (10x9)/(1*2), which reduces to 90/2, or 45). The combinations for three, four, five and six participants drawn from a set of ten is, respectively, 120, 210, 252, and 210. Monte Carlo sampling refers to the approach of random sampling with replacement from the set of all possible combinations. To achieve stability of estimates using this approach, the usual practice is to sample a fairly large number of times.

It is not difficult to program the computation of *p* using the factorial combination approach for small-scale distributions (such as the hypothetical distribution in Table 1). Following this approach for the more extensive distributions reported in the literature, though, can exhaust the resources of a personal computer. For example, the Mantel distribution (Nielsen & Molich, 1990) contains data from 76 participants and 30 discovered problems. The number of combinations for 76 participants taken six at a time is 218,618,940. As long as the number of iterations in a Monte Carlo program for estimating *p* is relatively modest (say, 1000 iterations), it can easily work within the limitations of modern personal computers.

The primary purpose of the current experiment is to estimate *p* from the hypothetical data in Table 1 and a variety of published sets of data using both factorial combination and Monte Carlo methods. If the estimates for *p* and associated statistics (standard deviation, interquartile range, etc.) from the two approaches are sufficiently similar, this would validate the Monte Carlo method for sets of data that are too resource-intensive for the use of factorial combination.

## Method

I wrote BASIC programs to estimate the following statistics from problem discovery databases using factorial combinations (see Appendix A for a sample program) and Monte Carlo estimation with 1000 iterations (see Appendix B for a sample program):

- mean value of *p*
- standard deviation of *p*
- root mean square error for estimated *p* against true *p*
- standard error of the mean for *p*
- delta for a 99% confidence interval around *p*
- upper and lower bounds for a 99% confidence interval around *p*
- $1^{st}$, $5^{th}$, $10^{th}$, $25^{th}$, $50^{th}$, $75^{th}$, $90^{th}$, $95^{th}$, and $99^{th}$ percentiles for the distribution of *p*

I ran these programs on a Micron Millennia[1] computer (Windows[2] 95, 64 MB memory) to evaluate the following published problem discovery databases for sample sizes ranging from two to six participants for the Monte Carlo method and for as many participants as possible using factorial combinations (in most cases, from two to four participants):

- MACERR (Lewis, 1994; Lewis, Henry, & Mack, 1990)
- VIRZI90 (Virzi, 1990; 1992)
- MANTEL (Nielsen & Molich, 1990)

In addition to the published databases, I also evaluated the hypothetical database provided in the introduction to this report (SAMPLE) and various subsets of published databases designed to give p for that subset a particular value. These subsets were:

- MACERR10: 45 problems selected to produce p=.10
- MACERR25: 34 problems selected to produce p=.25
- MACERR50: 10 problems selected to produce p=.50
- MACERR73: the three highest frequency problems with average p=.73

Tables in Appendix C document these problem discovery distributions.

To evaluate the effectiveness of the RANDOM function in BASIC for selecting participants for the Monte Carlo evaluations, the Monte Carlo program kept track of the frequency of selection of each participant. With this information, it is possible to calculate a $\chi^2$ statistic (Steele & Torrie, 1960) to determine if participant selection by the program was significantly nonrandom.

---

[1] Micron and Millennia are trademarks or registered trademarks of Micron Inc.
[2] Windows is a trademark or registered trademark of Microsoft Corp.

## Results

### Estimates of Mean *p*

Tables 4 and 5 show the means of the estimates of *p* for the various published databases and the selected subsets. OOM in a cell indicates that the personal computer was unable to run the program due to an out-of-memory condition. The values in the table show high correspondence between the output of the two programs, with no differences in the estimates of *p* (where available) exceeding 0.006. Table 6 shows the differences between factorial and Monte Carlo estimates for sample sizes from two to four. The average difference was 0.000.

*Table 4. Estimates of mean p for published databases*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.567 | 0.421 | 0.347 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.568 | 0.421 | 0.346 | 0.301 | 0.269 |
| *VIRZI90* | *Factorial* | *0.36* | 0.661 | 0.543 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.661 | 0.544 | 0.484 | 0.448 | 0.425 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.724 | 0.622 | 0.572 | 0.536 | 0.511 |

*Table 5. Estimates of mean p for hypothetical and subset data*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.521 | 0.357 | 0.275 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.523 | 0.357 | 0.276 | 0.225 | 0.193 |
| *MACERR25* | *Factorial* | *0.25* | 0.551 | 0.410 | 0.343 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.550 | 0.410 | 0.341 | 0.307 | 0.282 |
| *MACERR50* | *Factorial* | *0.50* | 0.651 | 0.556 | 0.520 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.645 | 0.558 | 0.519 | 0.509 | 0.502 |
| *MACERR73* | *Factorial* | *0.73* | 0.784 | 0.741 | 0.734 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.785 | 0.742 | 0.734 | 0.733 | 0.733 |
| *SAMPLE* | *Factorial* | *0.50* | 0.709 | 0.600 | 0.550 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.711 | 0.601 | 0.551 | 0.524 | 0.512 |

*Table 6. Difference scores for factorial and Monte Carlo estimates of mean p*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | -0.001 | 0.000 | 0.001 | |
| *VIRZI90* | *Difference* | *0.36* | 0.000 | -0.001 | OOM | |
| *MACERR10* | *Difference* | *0.10* | -0.002 | 0.000 | -0.001 | |
| *MACERR25* | *Difference* | *0.25* | 0.001 | 0.000 | 0.002 | |
| *MACERR50* | *Difference* | *0.50* | 0.006 | -0.002 | 0.001 | |
| *MACERR73* | *Difference* | *0.73* | -0.001 | -0.001 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | -0.002 | -0.001 | -0.001 | 0.000 |

### Estimates of the Standard Deviation of *p*

Estimates of variability such as the standard error of the mean and confidence interval bounds depend on accurate estimation of the standard deviation. Tables 7 and 8 show the standard deviations calculated for the published databases and selected subsets. Table 9 shows the differences between factorial and Monte Carlo estimates for sample sizes from two to four. The output of the two methods showed high correspondence, with an average difference of 0.000 and no difference exceeding 0.003.

*Table 7. Estimates of the standard deviation of p for published databases*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.033 | 0.026 | 0.022 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.032 | 0.027 | 0.021 | 0.018 | 0.015 |
| *VIRZI90* | *Factorial* | *0.36* | 0.047 | 0.039 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.046 | 0.040 | 0.032 | 0.028 | 0.026 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.063 | 0.060 | 0.053 | 0.045 | 0.044 |

*Table 8. Estimates of the standard deviation of p for hypothetical and subset data*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.052 | 0.034 | 0.022 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.055 | 0.032 | 0.025 | 0.016 | 0.012 |
| *MACERR25* | *Factorial* | *0.25* | 0.050 | 0.046 | 0.041 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.049 | 0.047 | 0.041 | 0.038 | 0.035 |
| *MACERR50* | *Factorial* | *0.50* | 0.093 | 0.085 | 0.078 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.095 | 0.086 | 0.076 | 0.069 | 0.062 |
| *MACERR73* | *Factorial* | *0.73* | 0.140 | 0.115 | 0.096 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.142 | 0.116 | 0.095 | 0.082 | 0.069 |
| *SAMPLE* | *Factorial* | *0.50* | 0.091 | 0.063 | 0.043 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.093 | 0.062 | 0.043 | 0.030 | 0.023 |

*Table 9. Difference scores for factorial and Monte Carlo estimates of standard deviation*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.001 | -0.001 | 0.001 | |
| *VIRZI90* | *Difference* | *0.36* | 0.001 | -0.001 | OOM | |
| *MACERR10* | *Difference* | *0.10* | -0.003 | 0.002 | -0.003 | |
| *MACERR25* | *Difference* | *0.25* | 0.001 | -0.001 | 0.000 | |
| *MACERR50* | *Difference* | *0.50* | -0.002 | -0.001 | 0.002 | |
| *MACERR73* | *Difference* | *0.73* | -0.002 | -0.001 | 0.001 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | -0.002 | 0.001 | 0.000 | 0.000 |

### Estimates of the Root Mean Square Error

The root mean square (rms) error is similar to a standard deviation, but rather than computing the mean squared difference between each data point and the mean of a distribution (the standard deviation), the computation is the mean squared difference between each data point and the true value of *p* for the distribution[3]. Tables 10 and 11 show the rms error calculated for the published databases and selected subsets. Table 12 shows the differences between factorial and Monte Carlo estimates for sample sizes from two to four. The output of the two methods showed high correspondence, with an average difference of 0.000 and no single difference exceeding 0.005.

*Table 10. Estimates of rms error for published databases*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.407 | 0.260 | 0.185 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.406 | 0.259 | 0.185 | 0.140 | 0.107 |
| *VIRZI90* | *Factorial* | *0.36* | 0.307 | 0.189 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.306 | 0.189 | 0.130 | 0.094 | 0.071 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.354 | 0.254 | 0.204 | 0.167 | 0.143 |

*Table 11. Estimates of rms error for hypothetical and subset data*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.424 | 0.257 | 0.174 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.425 | 0.257 | 0.175 | 0.124 | 0.092 |
| *MACERR25* | *Factorial* | *0.25* | 0.309 | 0.169 | 0.104 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.307 | 0.170 | 0.102 | 0.071 | 0.049 |
| *MACERR50* | *Factorial* | *0.50* | 0.178 | 0.102 | 0.081 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.173 | 0.104 | 0.079 | 0.070 | 0.063 |
| *MACERR73* | *Factorial* | *0.73* | 0.149 | 0.115 | 0.096 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.151 | 0.116 | 0.095 | 0.082 | 0.069 |
| *SAMPLE* | *Factorial* | *0.50* | 0.230 | 0.119 | 0.067 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.230 | 0.118 | 0.067 | 0.038 | 0.026 |

*Table 12. Difference scores for factorial and Monte Carlo estimates of rms error*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.001 | 0.001 | 0.000 | |
| *VIRZI90* | *Difference* | *0.36* | 0.001 | 0.000 | OOM | |
| *MACERR10* | *Difference* | *0.10* | -0.001 | 0.000 | -0.001 | |
| *MACERR25* | *Difference* | *0.25* | 0.002 | -0.001 | 0.002 | |
| *MACERR50* | *Difference* | *0.50* | 0.005 | -0.002 | 0.002 | |
| *MACERR73* | *Difference* | *0.73* | -0.002 | -0.001 | 0.001 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.001 | 0.000 | 0.000 |

[3] If not true *p*, using the best estimate of *p* available, computed from the full data set.

## Percentile Estimates

To assess estimation of the entire distribution, it is reasonable to compare certain key percentiles. Estimates of distributions tend to be most stable in the center, and least stable at the tails. The following set of tables (Tables 13 through 27) shows, for both published and subset/hypothetical distributions, the 50$^{th}$ percentile (median), 25$^{th}$ percentile, 75$^{th}$ percentile, 1$^{st}$ percentile and 99$^{th}$ percentile. Differences between factorial and Monte Carlo estimation of medians averaged 0.000, and no single difference was greater than .004. For 25$^{th}$ and 75$^{th}$ percentiles respectively, the mean differences were 0.001 and 0.000, with no difference exceeding 0.012. As expected, the tails (1$^{st}$ and 99$^{th}$ percentiles) were less stable, and matched up a little less precisely. For the 1$^{st}$ and 99$^{th}$ percentiles respectively, the mean differences were 0.005 and –0.004 (which is still quite consistent), with the largest single difference equal to 0.112.

*Table 13. Estimates of median p for published databases*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.565 | 0.422 | 0.347 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.565 | 0.422 | 0.346 | 0.301 | 0.269 |
| *VIRZI90* | *Factorial* | *0.36* | 0.659 | 0.540 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.658 | 0.543 | 0.483 | 0.447 | 0.424 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.727 | 0.621 | 0.574 | 0.538 | 0.508 |

*Table 14. Estimates of median p for hypothetical and subset data*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.500 | 0.352 | 0.272 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.500 | 0.354 | 0.271 | 0.224 | 0.192 |
| *MACERR25* | *Factorial* | *0.25* | 0.540 | 0.407 | 0.341 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.536 | 0.407 | 0.339 | 0.303 | 0.282 |
| *MACERR50* | *Factorial* | *0.50* | 0.643 | 0.556 | 0.525 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.643 | 0.556 | 0.525 | 0.500 | 0.500 |
| *MACERR73* | *Factorial* | *0.73* | 0.833 | 0.778 | 0.750 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.833 | 0.778 | 0.750 | 0.733 | 0.722 |
| *SAMPLE* | *Factorial* | *0.50* | 0.714 | 0.593 | 0.550 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.714 | 0.593 | 0.550 | 0.520 | 0.517 |

*Table 15. Difference scores for factorial and Monte Carlo estimates of median p*

| *Source* | Style | True p | Sample=2 | Sample=3 | Sample=4 | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.000 | 0.000 | 0.001 | |
| *VIRZI90* | *Difference* | *0.36* | 0.001 | -0.003 | OOM | |
| *MACERR10* | *Difference* | *0.10* | 0.000 | -0.002 | 0.001 | |
| *MACERR25* | *Difference* | *0.25* | 0.004 | 0.000 | 0.002 | |
| *MACERR50* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | |
| *MACERR73* | *Difference* | *0.73* | 0.000 | 0.000 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | 0.000 |

*Table 16. Estimates of the 25th percentile for published databases*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.539 | 0.403 | 0.332 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.544 | 0.401 | 0.332 | 0.290 | 0.258 |
| *VIRZI90* | *Factorial* | *0.36* | 0.630 | 0.519 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.630 | 0.519 | 0.461 | 0.429 | 0.407 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.679 | 0.583 | 0.536 | 0.504 | 0.480 |

*Table 17. Estimates of the 25th percentile for hypothetical and subset data*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.500 | 0.333 | 0.263 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.500 | 0.333 | 0.263 | 0.215 | 0.185 |
| *MACERR25* | *Factorial* | *0.25* | 0.500 | 0.375 | 0.313 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.500 | 0.375 | 0.310 | 0.281 | 0.258 |
| *MACERR50* | *Factorial* | *0.50* | 0.583 | 0.500 | 0.472 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.571 | 0.500 | 0.472 | 0.460 | 0.467 |
| *MACERR73* | *Factorial* | *0.73* | 0.667 | 0.667 | 0.667 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.667 | 0.667 | 0.667 | 0.667 | 0.667 |
| *SAMPLE* | *Factorial* | *0.50* | 0.643 | 0.556 | 0.525 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.643 | 0.556 | 0.525 | 0.500 | 0.500 |

*Table 18. Difference scores for factorial and Monte Carlo estimates of the 25th percentile*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | -0.005 | 0.002 | 0.000 | |
| *VIRZI90* | *Difference* | *0.36* | 0.000 | 0.000 | OOM | |
| *MACERR10* | *Difference* | *0.10* | 0.000 | 0.000 | 0.000 | |
| *MACERR25* | *Difference* | *0.25* | 0.000 | 0.000 | 0.003 | |
| *MACERR50* | *Difference* | *0.50* | 0.012 | 0.000 | 0.000 | |
| *MACERR73* | *Difference* | *0.73* | 0.000 | 0.000 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | 0.001 |

*Table 19. Estimates of the 75th percentile for published databases*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.593 | 0.441 | 0.362 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.588 | 0.441 | 0.360 | 0.313 | 0.280 |
| *VIRZI90* | *Factorial* | *0.36* | 0.690 | 0.568 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.688 | 0.568 | 0.500 | 0.467 | 0.443 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.767 | 0.667 | 0.605 | 0.567 | 0.540 |

*Table 20. Estimates of the 75th percentile for hypothetical and subset data*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.500 | 0.370 | 0.283 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.500 | 0.370 | 0.283 | 0.233 | 0.200 |
| *MACERR25* | *Factorial* | *0.25* | 0.579 | 0.440 | 0.370 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.583 | 0.440 | 0.369 | 0.333 | 0.303 |
| *MACERR50* | *Factorial* | *0.50* | 0.722 | 0.625 | 0.575 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.714 | 0.630 | 0.575 | 0.560 | 0.537 |
| *MACERR73* | *Factorial* | *0.73* | 0.833 | 0.778 | 0.833 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.833 | 0.778 | 0.833 | 0.800 | 0.778 |
| *SAMPLE* | *Factorial* | *0.50* | 0.750 | 0.625 | 0.583 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.750 | 0.625 | 0.583 | 0.540 | 0.533 |

*Table 21. Difference scores for factorial and Monte Carlo estimates of the 75th percentile*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.005 | 0.000 | 0.002 | |
| *VIRZI90* | *Difference* | *0.36* | 0.002 | 0.000 | OOM | |
| *MACERR10* | *Difference* | *0.10* | 0.000 | 0.000 | 0.000 | |
| *MACERR25* | *Difference* | *0.25* | -0.004 | 0.000 | 0.001 | |
| *MACERR50* | *Difference* | *0.50* | 0.008 | -0.005 | 0.000 | |
| *MACERR73* | *Difference* | *0.73* | 0.000 | 0.000 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | 0.000 |

*Table 22. Estimates of the 1st percentile for published databases*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.500 | 0.362 | 0.297 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.500 | 0.364 | 0.298 | 0.260 | 0.231 |
| *VIRZI90* | *Factorial* | *0.36* | 0.563 | 0.462 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.563 | 0.464 | 0.414 | 0.389 | 0.369 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.571 | 0.487 | 0.450 | 0.429 | 0.417 |

*Table 23. Estimates of the 1st percentile for hypothetical and subset data*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | Sample=5 | Sample=6 |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.500 | 0.333 | 0.250 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.500 | 0.333 | 0.250 | 0.200 | 0.167 |
| *MACERR25* | *Factorial* | *0.25* | 0.500 | 0.333 | 0.260 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.500 | 0.333 | 0.250 | 0.227 | 0.205 |
| *MACERR50* | *Factorial* | *0.50* | 0.500 | 0.375 | 0.350 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.500 | 0.375 | 0.361 | 0.356 | 0.367 |
| *MACERR73* | *Factorial* | *0.73* | 0.500 | 0.556 | 0.500 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 0.500 | 0.444 | 0.500 | 0.533 | 0.556 |
| *SAMPLE* | *Factorial* | *0.50* | 0.550 | 0.500 | 0.475 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.550 | 0.500 | 0.475 | 0.460 | 0.467 |

*Table 24. Difference scores for factorial and Monte Carlo estimates of the 1st percentile*

| Source | Style | True p | Sample=2 | Sample=3 | Sample=4 | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.000 | -0.002 | -0.001 | |
| *VIRZI90* | *Difference* | *0.36* | 0.000 | -0.002 | OOM | |
| *MACERR10* | *Difference* | *0.10* | 0.000 | 0.000 | 0.000 | |
| *MACERR25* | *Difference* | *0.25* | 0.000 | 0.000 | 0.010 | |
| *MACERR50* | *Difference* | *0.50* | 0.000 | 0.000 | -0.011 | |
| *MACERR73* | *Difference* | *0.73* | 0.000 | 0.112 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | 0.005 |

*Table 25. Estimates of the 99th percentile for published databases*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR* | *Factorial* | *0.16* | 0.646 | 0.475 | 0.394 | OOM | OOM |
| *MACERR* | *Monte Carlo* | *0.16* | 0.646 | 0.475 | 0.390 | 0.340 | 0.303 |
| *VIRZI90* | *Factorial* | *0.36* | 0.800 | 0.639 | OOM | OOM | OOM |
| *VIRZI90* | *Monte Carlo* | *0.36* | 0.773 | 0.640 | 0.565 | 0.526 | 0.494 |
| *MANTEL* | *Factorial* | *0.38* | OOM | OOM | OOM | OOM | OOM |
| *MANTEL* | *Monte Carlo* | *0.38* | 0.875 | 0.769 | 0.702 | 0.643 | 0.615 |

*Table 26. Estimates of the 99th percentile for hypothetical and subset data*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|---|
| *MACERR10* | *Factorial* | *0.10* | 0.750 | 0.500 | 0.357 | OOM | OOM |
| *MACERR10* | *Monte Carlo* | *0.10* | 0.750 | 0.500 | 0.375 | 0.267 | 0.227 |
| *MACERR25* | *Factorial* | *0.25* | 0.684 | 0.533 | 0.446 | OOM | OOM |
| *MACERR25* | *Monte Carlo* | *0.25* | 0.684 | 0.536 | 0.435 | 0.394 | 0.374 |
| *MACERR50* | *Factorial* | *0.50* | 0.833 | 0.741 | 0.694 | OOM | OOM |
| *MACERR50* | *Monte Carlo* | *0.50* | 0.917 | 0.741 | 0.700 | 0.660 | 0.650 |
| *MACERR73* | *Factorial* | *0.73* | 1.000 | 1.000 | 0.917 | OOM | OOM |
| *MACERR73* | *Monte Carlo* | *0.73* | 1.000 | 1.000 | 0.917 | 0.933 | 0.889 |
| *SAMPLE* | *Factorial* | *0.50* | 0.917 | 0.762 | 0.656 | OOM | OOM |
| *SAMPLE* | *Monte Carlo* | *0.50* | 0.917 | 0.762 | 0.656 | 0.600 | 0.574 |

*Table 27. Difference scores for factorial and Monte Carlo estimates of the 99th percentile*

| ***Source*** | **Style** | **True p** | **Sample=2** | **Sample=3** | **Sample=4** | |
|---|---|---|---|---|---|---|
| *MACERR* | *Difference* | *0.16* | 0.000 | 0.000 | 0.004 | |
| *VIRZI90* | *Difference* | *0.36* | 0.027 | -0.001 | OOM | |
| *MACERR10* | *Difference* | *0.10* | 0.000 | 0.000 | -0.018 | |
| *MACERR25* | *Difference* | *0.25* | 0.000 | -0.003 | 0.011 | |
| *MACERR50* | *Difference* | *0.50* | -0.084 | 0.000 | -0.006 | |
| *MACERR73* | *Difference* | *0.73* | 0.000 | 0.000 | 0.000 | **Average** |
| *SAMPLE* | *Difference* | *0.50* | 0.000 | 0.000 | 0.000 | -0.004 |

## Adequacy of Random Sampling of Participants from Problem-Discovery Databases

For certain types of sampling, the random functions of programming languages are not sufficiently random. To check the adequacy of random sampling of participants from the problem-discovery databases with the Monte Carlo program, I conducted $\chi^2$ analyses for each combination of source database and sample size. Table 28 contains the results of these 40 analyses (eight sources by five sample sizes). Using a criterion of $\alpha$=.05 to reject the hypothesis of random sampling, the expected number of rejections by chance across 40 opportunities is 40(.05), or 2.0. The observed number of rejections was 1 (SAMPLE with a sample size of 3), well within expectation, indicating adequately random sampling by the Monte Carlo program.

*Table 28. Results of $c^2$ tests to check adequacy of random sampling*

| ***Source*** | ***Statistic*** | **Sample=2** | **Sample=3** | **Sample=4** | **Sample=5** | **Sample=6** |
|---|---|---|---|---|---|---|
| *MACERR* | $c^2$ | 14.8 | 5.8 | 6.8 | 6.6 | 11.0 |
| | *df* | 14 | 14 | 14 | 14 | 14 |
| | *p* | 0.39 | 0.97 | 0.94 | 0.95 | 0.68 |
| *VIRZI* | $c^2$ | 14.9 | 17.2 | 15.7 | 13.8 | 14.4 |
| | *df* | 19 | 19 | 19 | 19 | 19 |
| | *p* | 0.73 | 0.57 | 0.68 | 0.80 | 0.76 |
| *MANTEL* | $c^2$ | 80.4 | 81.1 | 61.5 | 63.0 | 79.8 |
| | *df* | 75 | 75 | 75 | 75 | 75 |
| | *p* | 0.31 | 0.30 | 0.87 | 0.84 | 0.33 |
| *MACERR10* | $c^2$ | 8.5 | 8.9 | 17.9 | 12.9 | 8.1 |
| | *df* | 14 | 14 | 14 | 14 | 14 |
| | *p* | 0.86 | 0.84 | 0.21 | 0.53 | 0.89 |
| *MACERR25* | $c^2$ | 18.8 | 6.0 | 13.4 | 8.2 | 4.7 |
| | *df* | 14 | 14 | 14 | 14 | 14 |
| | *p* | 0.17 | 0.97 | 0.49 | 0.88 | 0.99 |
| *MACERR50* | $c^2$ | 14.1 | 7.5 | 8.3 | 6.3 | 7.7 |
| | *df* | 14 | 14 | 14 | 14 | 14 |
| | *p* | 0.45 | 0.91 | 0.87 | 0.96 | 0.90 |
| *MACERR73* | $c^2$ | 8.5 | 11.7 | 6.4 | 9.4 | 6.5 |
| | *df* | 14 | 14 | 14 | 14 | 14 |
| | *p* | 0.86 | 0.63 | 0.96 | 0.80 | 0.95 |
| *SAMPLE* | $c^2$ | 14.1 | 17.1 | 8.2 | 5.9 | 1.9 |
| | *df* | 9 | 9 | 9 | 9 | 9 |
| | *p* | 0.12 | 0.05 | 0.51 | 0.75 | 0.99 |

## Discussion

The results clearly validate the use of the Monte Carlo program for future investigations of the properties of problem-discovery *p*. For the essential measures of mean *p*, standard deviation of *p*, and root mean square error, median, and 25th and 75th percentiles, the outputs of the factorial and Monte Carlo programs were essentially identical. Even at the extreme tails of the distributions of p (the 1st and 99th percentiles), the outputs of the programs were extremely close. The results of the $\chi^2$ analyses – demonstrating adequately random sampling across participants in the problem discovery databases – also support the use of the Monte Carlo program.

## References


Diamond, W. J. (1981). *Practical experiment designs for engineers and scientists.* Belmont, CA: Lifetime Learning Publications.

Hertzum, M., & Jacobsen, N. (In press). The evaluator effect in usability evaluation methods: A chilling fact about a burning issue. To appear in *The International Journal of Human-Computer Interaction*.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *In Conference Proceedings on Human Factors in Computing Systems – CHI90* (pp. 249-256). New York, NY: ACM.

Lewis, J. R. (1991). *Legitimate use of small sample sizes in usability studies: Three examples* (Tech. Report 54.594). Boca Raton, FL: International Business Machines Corp.

Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors*, *36*, 368-378.

Lewis, J. R. (2000). *Sample size estimation and use of substitute audiences* (Tech. report in press). Raleigh, NC: International Business Machines Corp.

Lewis, J. R., Henry, S. C., & Mack, R. L. (1990). Integrated office scenario benchmarks: A case study. In *Human Computer Interaction -- INTERACT '90* (pp. 337-343). Cambridge, England: Elsevier Science Publishers, IFIP.

Steele, R. G. D., & Torrie, J. H. (1960). *Principles and procedures of statistics*. New York, NY: McGraw-Hill.

Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. In *Proceedings of the Human Factors Society 34th Annual Meeting* (pp. 291-294). Santa Monica, CA: Human Factors Society.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, *34*, 443-451.

Walpole, R. E. (1976). *Elementary statistical concepts.* New York, NY: Macmillan.

# Appendix A. Sample Program Using Factorial Combination

```
10 'Program to do systematic simulation for adjustment of p  (J. R. Lewis 7/04/00)
20 '
30 'The program expects to find the errors in rows separated by commas (cols are participants).
40 'The first column is the problem number.
50 'The last column is reserved for an impact rating.
60 'This version of the problem deals with participant subsets of size 4.
70 '
80 CLS
90 'INPUT "Filename for error data: ",F$
100 F$="macerr10.asc"
110 'input "Filename for output: ",ot$
120 OT$="p4mac10.csv"
130 OPEN F$ FOR INPUT AS #1
140 'INPUT "Number of identified problems/errors: ",E
150 INPUT #1, E
160 'INPUT "Number of participants: ",P
170 INPUT #1, P
180 'input "Size of participant subset: ",o
190 O=4 'Do not change this number!
200 'Number of combinations of p items taken o at a time
210 LET NUM=P:LET DEN=O
220 FOR X=1 TO O-1
230 LET NUM=NUM*(P-X):LET DEN=DEN*(O-X)
240 NEXT X
250 LET C=NUM/DEN
251 'Calculate indices for key percentiles
252 for x=1 to c
253 let perc=x*100/c
254 if pin1fnd=1 then goto 256
255 if perc>=1 then pin1=x:pin1fnd=1
256 if pin5fnd=1 then goto 258
257 if perc>=5 then pin5=x:pin5fnd=1
258 if pin10fnd=1 then goto 260
259 if perc>=10 then pin10=x:pin10fnd=1
260 if pin25fnd=1 then goto 262
261 if perc>=25 then pin25=x:pin25fnd=1
262 if pin50fnd=1 then goto 264
263 if perc>=50 then pin50=x:pin50fnd=1
264 if pin75fnd=1 then goto 266
265 if perc>=75 then pin75=x:pin75fnd=1
266 if pin90fnd=1 then goto 268
267 if perc>=90 then pin90=x:pin90fnd=1
268 if pin95fnd=1 then goto 270
269 if perc>=95 then pin95=x:pin95fnd=1
270 if pin99fnd=1 then goto 272
271 if perc>=99 then pin99=x:pin99fnd=1
272 next x
273 'print pin1,pin5,pin10,pin25,pin50,pin75,pin90,pin95,pin99:input junk
278 DIM ERRORS(E,P), ERRCNT(E), EST(C), ADJ1(C), ADJ2(C), ADJ3(C), ADJ4(C)
279 'Read in error matrix -- first column has err num, last column can hold impact rating
280 FOR X=1 TO E
290 INPUT #1, JUNK 'discard error id number
300 FOR Y=1 TO P 'get error occurrence for each participant
310 INPUT #1, ERRORS(X,Y)
320 NEXT Y
330 INPUT #1, JUNK 'discard impact rating
340 NEXT X
350 CLOSE #1
360 PRINT "Table of Problem Occurrence"
370 FOR X=1 TO E
380 FOR Y=1 TO P
390 PRINT ERRORS(X,Y);" ";
400 NEXT Y
410 PRINT
420 NEXT X
430 PRINT:PRINT
440 'Establish combinations, calculate and write data
450 OPEN OT$ FOR OUTPUT AS #2
460 PRINT "Errdat Table":PRINT
470 PRINT #2,"Errdat Table":PRINT #2,""
475 for w=1 to p-3
480 FOR X=w+1 TO P-2
490 FOR Y=X+1 TO P-1
500 FOR Z=Y+1 TO P
510 LET CNT=CNT+1
520 FOR Q=1 TO E
530 LET ERRCNT(Q)=errors(q,w)+ERRORS(Q,X)+ERRORS(Q,Y)+ERRORS(Q,Z)
540 NEXT Q
550 PRINT w;" ";X;" ";Y;" ";Z;" ":FOR Q=1 TO e:PRINT ERRCNT(Q);" ";:NEXT Q
560 'Find total number of problems for this subset
570 LET PZERO=0 'initialize pzero -- the number of problem counts equal to 0
580 LET PONE=0 'initialize pone -- the number of problem counts equal to 1
590 FOR Q=1 TO E
600 IF ERRCNT(Q)=0 THEN LET PZERO=PZERO+1
610 IF ERRCNT(Q)=1 THEN LET PONE=PONE+1
620 NEXT Q
630 LET TPROB=E-PZERO  'tprob is total number of discovered problems for this subset
640 'Find total number of problem occurrences
650 LET TOCC=0 'initialize tocc -- total number of problem occurrences for this subset
660 FOR Q=1 TO E
670 LET TOCC=TOCC+ERRCNT(Q)
680 NEXT Q
690 LET TOPP=TPROB*O 'Total number of opportunities for error
700 LET ESTP=TOCC/TOPP 'Estimate of p (unadjusted)
710 LET ADJOPP1=TOPP+PONE 'Adjust opportunities with pone
720 LET ADJP1=TOCC/ADJOPP1 'Adjusted estimate of p with pone
730 LET ADJOPP2=TOPP+TPROB 'Adjust opportunities with tprob
740 LET ADJP2=TOCC/ADJOPP2 'Adjusted estimate of p with tocc
750 LET ADJOPP3=TOPP+TOCC 'Adjust opportunities with tocc
760 LET ADJP3=TOCC/ADJOPP3 'Adjusted estimate of p with tocc
770 LET ADJP4=ESTP*ADJP2+(1-ESTP)*ADJP3 'Linear interp of 2 and 3 using estp for weight
780 PRINT CNT;" ";X;" ";Y;" ";Z;" ";
```

```
790 PRINT #2,CNT;",";X;",";Y;",";Z;",";
800 FOR Q=1 TO E
810 PRINT ERRCNT(Q);" ";
820 PRINT #2,ERRCNT(Q);",";
830 NEXT Q
840 PRINT TPROB;" ";TOCC;" ";TOPP;" ";ESTP;" ";ADJP1;" ";ADJP2;" ";ADJP3;" ";ADJP4
850 PRINT #2, TPROB;",";TOCC;",";TOPP;",";ESTP;",";ADJP1;",";ADJP2;",";ADJP3;",";ADJP4
860 LET SUMESTP=SUMESTP+ESTP:LET SUMADJP1=SUMADJP1+ADJP1
870 LET SUMADJP2=SUMADJP2+ADJP2:LET SUMADJP3=SUMADJP3+ADJP3:LET SUMADJP4=SUMADJP4+ADJP4
880 LET EST(CNT)=ESTP:LET ADJ1(CNT)=ADJP1:LET ADJ2(CNT)=ADJP2
890 LET ADJ3(CNT)=ADJP3:LET ADJ4(CNT)=ADJP4
900 NEXT Z
901 NEXT Y
902 NEXT X
903 next w
913 'Shell sort the p arrays
914 PRINT:PRINT "Sorting est"
915 let n=c
916 'for x=1 to n:print est(x);" ";:next x:print "906-est-unsorted":input junk
920 M8=n
930 M8=INT(M8/2)
940 IF M8=0 THEN 1040
950 K8=N-M8 : J8=1
960 I8=J8
970 L8=I8+M8
980 IF est(I8)<=est(L8) THEN 1020
990 T8=est(I8) : est(I8)=est(L8)
1000 est(L8)=T8 : I8=I8-M8
1010 IF I8>=1 THEN 970
1020 J8=J8+1
1030 IF J8<=K8 THEN 960 ELSE 930
1040 'for x=1 to n:print est(x);" ";:next x:print "1040-est-sorted":input junk
1071 PRINT:PRINT "Sorting adj1"
1110 'for x=1 to n:print adj1(x);" ";:next x:print "1110-adj1-unsorted":input junk
1120 M8=n
1130 M8=INT(M8/2)
1140 IF M8=0 THEN 1240
1150 K8=N-M8 : J8=1
1160 I8=J8
1170 L8=I8+M8
1180 IF adj1(I8)<=adj1(L8) THEN 1220
1190 T8=adj1(I8) : adj1(I8)=adj1(L8)
1200 adj1(L8)=T8 : I8=I8-M8
1210 IF I8>=1 THEN 1170
1220 J8=J8+1
1230 IF J8<=K8 THEN 1160 ELSE 1130
1240 'for x=1 to n:print adj1(x);" ";:next x:print "1240-adj1-sorted":input junk
1241 PRINT:PRINT "Sorting adj2"
1242 'for x=1 to c:print adj2(x);" ";:next x:print "1242-adj2-unsorted":input junk
1243 n=0 'checking that changing n to c will fix sort
1245 M8=C
1246 M8=INT(M8/2)
1247 IF M8=0 THEN goto 1330
1248 K8=c-M8 : J8=1
1250 I8=J8
1260 L8=I8+M8
1270 IF ADJ2(I8)<=ADJ2(L8) THEN 1310
1280 T8=ADJ2(I8) : ADJ2(I8)=ADJ2(L8)
1290 ADJ2(L8)=T8 : I8=I8-M8
1300 IF I8>=1 THEN 1260
1310 J8=J8+1
1320 IF J8<=K8 THEN 1250 ELSE 1246
1330 'for x=1 to c:print adj2(x);" ";:next x:print "1330-adj2-sorted":input junk
1331 PRINT:PRINT "Sorting adj3"
1333 'for x=1 to c:print adj3(x);" ";:next x:print "1333-adj3-unsorted":input junk
1340 M8=C
1350 M8=INT(M8/2)
1360 IF M8=0 THEN goto 1460
1370 K8=c-M8 : J8=1
1380 I8=J8
1390 L8=I8+M8
1400 IF ADJ3(I8)<=ADJ3(L8) THEN 1440
1410 T8=ADJ3(I8) : ADJ3(I8)=ADJ3(L8)
1420 ADJ3(L8)=T8 : I8=I8-M8
1430 IF I8>=1 THEN 1390
1440 J8=J8+1
1450 IF J8<=K8 THEN 1380 ELSE 1350
1460 'for x=1 to c:print adj3(x);" ";:next x:print "1460-adj3-sorted":input junk
1461 PRINT:PRINT "Sorting adj4"
1462 'for x=1 to c:print adj4(x);" ";:next x:print "1462-adj4-unsorted":input junk
1470 M8=C
1480 M8=INT(M8/2)
1490 IF M8=0 THEN goto 1590
1500 K8=c-M8 : J8=1
1510 I8=J8
1520 L8=I8+M8
1530 IF ADJ4(I8)<=ADJ4(L8) THEN 1570
1540 T8=ADJ4(I8) : ADJ4(I8)=ADJ4(L8)
1550 ADJ4(L8)=T8 : I8=I8-M8
1560 IF I8>=1 THEN 1520
1570 J8=J8+1
1580 IF J8<=K8 THEN 1510 ELSE 1480
1590 'for x=1 to c:print adj4(x);" ";:next x:print "1590-adj4-sorted":input junk
1591 'Calculate final statistics
1600 FOR X=1 TO E
1610 FOR Y=1 TO P
1620 LET GRANDERR=GRANDERR+ERRORS(X,Y)
1630 LET GRANDOPP=GRANDOPP+1
1640 NEXT Y
1650 NEXT X
1660 LET TRUEP=GRANDERR/GRANDOPP
1670 LET ESTP=SUMESTP/C:LET ADJP1=SUMADJP1/C
1680 LET ADJP2=SUMADJP2/C:LET ADJP3=SUMADJP3/C:LET ADJP4=SUMADJP4/C
1690 FOR X=1 TO C
1700 LET VARESTP=VARESTP+(EST(X)-ESTP)^2
1710 LET VARADJP1=VARADJP1+(ADJ1(X)-ADJP1)^2
1720 LET VARADJP2=VARADJP2+(ADJ2(X)-ADJP2)^2
1730 LET VARADJP3=VARADJP3+(ADJ3(X)-ADJP3)^2
```

```
1740 LET VARADJP4=VARADJP4+(ADJ4(X)-ADJP4)^2
1745 LET RMSestp=RMSestp+(est(X)-truep)^2
1750 LET RMSADJP1=RMSADJP1+(ADJ1(X)-truep)^2
1760 LET RMSADJP2=RMSADJP2+(ADJ2(X)-truep)^2
1770 LET RMSADJP3=RMSADJP3+(ADJ3(X)-truep)^2
1780 LET RMSADJP4=RMSADJP4+(ADJ4(X)-truep)^2
1790 NEXT X
1800 LET VARESTP=VARESTP/(C-1):LET VARADJP1=VARADJP1/(C-1)
1810 LET VARADJP2=VARADJP2/(C-1):LET VARADJP3=VARADJP3/(C-1):LET VARADJP4=VARADJP4/(C-1)
1820 LET SDESTP=VARESTP^.5:LET SDADJP1=VARADJP1^.5
1830 LET SDADJP2=VARADJP2^.5:LET SDADJP3=VARADJP3^.5:LET SDADJP4=VARADJP4^.5
1840 LET SEMESTP=SDESTP/(C^.5):LET SEMADJP1=SDADJP1/(C^.5)
1850 LET SEMADJP2=SDADJP2/(C^.5):LET SEMADJP3=SDADJP3/(C^.5):LET SEMADJP4=SDADJP4/(C^.5)
1860 LET Z99=2.576
1870 LET D99ESTP=Z99*SEMESTP:LET D99ADJP1=Z99*SEMADJP1
1880 LET D99ADJP2=Z99*SEMADJP2:LET D99ADJP3=Z99*SEMADJP3:LET D99ADJP4=Z99*SEMADJP4
1890 LET UP99EST=ESTP+D99ESTP:LET UP99ADJ1=ADJP1+D99ADJP1
1900 LET UP99ADJ2=ADJP2+D99ADJP2:LET UP99ADJ3=ADJP3+D99ADJP3:LET UP99ADJ4=ADJP4+D99ADJP4
1910 LET LOW99EST=ESTP-D99ESTP:LET LOW99ADJ1=ADJP1-D99ADJP1
1920 LET LOW99ADJ2=ADJP2-D99ADJP2:LET LOW99ADJ3=ADJP3-D99ADJP3:LET LOW99ADJ4=ADJP4-D99ADJP4
1930 'print stats
1940 CLS
1950 PRINT "Statistic ","Original ";"Adj-Ones ";"Adj-Prbs ";"Adj-Occs ";"Adj-LinOp"
1960 PRINT #2,"Statistic";",";"Original";",";"Adj-Ones";",";"Adj-Probs";",";"Adj-Occs";",";
1970 PRINT #2,"Adj-LinOp"
1980 PRINT "--------- ","-------- ";"-------- ";"-------- ";"-------- ";"-------- "
1990 PRINT "Mean",
2000 PRINT #2,"Mean";",";
2010 PRINT USING "  #.###  ";ESTP,ADJP1,ADJP2,ADJP3,ADJP4
2020 PRINT #2, USING "#.###";ESTP;:PRINT #2,",";
2030 PRINT #2, USING "#.###";ADJP1;:PRINT #2,",";
2040 PRINT #2, USING "#.###";ADJP2;:PRINT #2,",";
2050 PRINT #2, USING "#.###";ADJP3;:PRINT #2,",";
2060 PRINT #2, USING "#.###";ADJP4
2070 PRINT "Std Dev",
2080 PRINT #2,"Std Dev";",";
2090 PRINT USING "  #.###  ";SDESTP,SDADJP1,SDADJP2,SDADJP3,SDADJP4
2100 PRINT #2, USING "#.###";SDESTP;:PRINT #2,",";
2110 PRINT #2, USING "#.###";SDADJP1;:PRINT #2,",";
2120 PRINT #2, USING "#.###";SDADJP2;:PRINT #2,",";
2130 PRINT #2, USING "#.###";SDADJP3;:PRINT #2,",";
2140 PRINT #2, USING "#.###";SDADJP4
2150 PRINT "RMS Err",
2160 PRINT #2,"RMS Err";",";
2161 let rmsadjp1=((rmsadjp1/(c-1))^.5)
2162 let rmsadjp2=((rmsadjp2/(c-1))^.5)
2163 let rmsadjp3=((rmsadjp3/(c-1))^.5)
2164 let rmsadjp4=((rmsadjp4/(c-1))^.5)
2165 let rmsestp=((rmsestp/(c-1))^.5)
2170 PRINT USING "  #.###  ";rmsESTP,RMSADJP1,RMSADJP2,RMSADJP3,RMSADJP4
2180 PRINT #2, USING "#.###";rmsESTP;:PRINT #2,",";
2190 PRINT #2, USING "#.###";RMSADJP1;:PRINT #2,",";
2200 PRINT #2, USING "#.###";RMSADJP2;:PRINT #2,",";
2210 PRINT #2, USING "#.###";RMSADJP3;:PRINT #2,",";
2220 PRINT #2, USING "#.###";RMSADJP4
2230 PRINT "sem",
2240 PRINT #2, "sem";",";
2250 PRINT USING "  #.###  ";SEMESTP,SEMADJP1,SEMADJP2,SEMADJP3,SEMADJP4
2260 PRINT #2, USING "#.###";SEMESTP;:PRINT #2,",";
2270 PRINT #2, USING "#.###";SEMADJP1;:PRINT #2,",";
2280 PRINT #2, USING "#.###";SEMADJP2;:PRINT #2,",";
2290 PRINT #2, USING "#.###";SEMADJP3;:PRINT #2,",";
2300 PRINT #2, USING "#.###";SEMADJP4
2310 PRINT "d99",
2320 PRINT #2, "d99";",";
2330 PRINT USING "  #.###  ";D99ESTP,D99ADJP1,D99ADJP2,D99ADJP3,D99ADJP4
2340 PRINT #2, USING "#.###";D99ESTP;:PRINT #2,",";
2350 PRINT #2, USING "#.###";D99ADJP1;:PRINT #2,",";
2360 PRINT #2, USING "#.###";D99ADJP2;:PRINT #2,",";
2370 PRINT #2, USING "#.###";D99ADJP3;:PRINT #2,",";
2380 PRINT #2, USING "#.###";D99ADJP4
2390 'PRINT
2400 PRINT #2,""
2410 PRINT "Upper",
2420 PRINT #2, "Upper";",";
2430 PRINT USING "  #.###  ";UP99EST,UP99ADJ1,UP99ADJ2,UP99ADJ3,UP99ADJ4
2440 PRINT #2, USING "#.###";UP99EST;:PRINT #2,",";
2450 PRINT #2, USING "#.###";UP99ADJ1;:PRINT #2,",";
2460 PRINT #2, USING "#.###";UP99ADJ2;:PRINT #2,",";
2470 PRINT #2, USING "#.###";UP99ADJ3;:PRINT #2,",";
2480 PRINT #2, USING "#.###";UP99ADJ4
2490 PRINT "Mean",
2500 PRINT #2, "Mean";",";
2510 PRINT USING "  #.###  ";ESTP,ADJP1,ADJP2,ADJP3,ADJP4
2520 PRINT #2, USING "#.###";ESTP;:PRINT #2,",";
2530 PRINT #2, USING "#.###";ADJP1;:PRINT #2,",";
2540 PRINT #2, USING "#.###";ADJP2;:PRINT #2,",";
2550 PRINT #2, USING "#.###";ADJP3;:PRINT #2,",";
2560 PRINT #2, USING "#.###";ADJP4
2570 PRINT "Lower",
2580 PRINT #2, "Lower";",";
2590 PRINT USING "  #.###  ";LOW99EST,LOW99ADJ1,LOW99ADJ2,LOW99ADJ3,LOW99ADJ4
2600 PRINT #2, USING "#.###";LOW99EST;:PRINT #2,",";
2610 PRINT #2, USING "#.###";LOW99ADJ1;:PRINT #2,",";
2620 PRINT #2, USING "#.###";LOW99ADJ2;:PRINT #2,",";
2630 PRINT #2, USING "#.###";LOW99ADJ3;:PRINT #2,",";
2640 PRINT #2, USING "#.###";LOW99ADJ4
2650 PRINT:PRINT
2660 PRINT #2, "":PRINT #2,""
2700 'print percentiles
2703 PRINT "1st %ile",
2706 print #2,"1st %ile";",";
2710 print using "  #.###  ";est(pin1),adj1(pin1),adj2(pin1),adj3(pin1),adj4(pin1)
2711 PRINT #2, USING "#.###";EST(pin1);:PRINT #2,",";
2712 PRINT #2, USING "#.###";adj1(pin1);:PRINT #2,",";
2720 PRINT #2, USING "#.###";adj2(pin1);:PRINT #2,",";
2730 PRINT #2, USING "#.###";ADJ3(pin1);:PRINT #2,",";
2740 PRINT #2, USING "#.###";ADJ4(pin1)
```

```
2803 PRINT "5th %ile",
2806 print #2,"5th %ile";",";
2810 print using "  #.###  ";est(pin5),adj1(pin5),adj2(pin5),adj3(pin5),adj4(pin5)
2811 PRINT #2, USING "#.###";EST(pin5);:PRINT #2,",";
2812 PRINT #2, USING "#.###";adj1(pin5);:PRINT #2,",";
2820 PRINT #2, USING "#.###";adj2(pin5);:PRINT #2,",";
2830 PRINT #2, USING "#.###";ADJ3(pin5);:PRINT #2,",";
2840 PRINT #2, USING "#.###";ADJ4(pin5)
2903 PRINT "10th %ile",
2906 print #2,"10th %ile";",";
2910 print using "  #.###  ";est(pin10),adj1(pin10),adj2(pin10),adj3(pin10),adj4(pin10)
2911 PRINT #2, USING "#.###";EST(pin10);:PRINT #2,",";
2912 PRINT #2, USING "#.###";adj1(pin10);:PRINT #2,",";
2920 PRINT #2, USING "#.###";adj2(pin10);:PRINT #2,",";
2930 PRINT #2, USING "#.###";ADJ3(pin10);:PRINT #2,",";
2940 PRINT #2, USING "#.###";ADJ4(pin10)
3003 PRINT "25th %ile",
3006 print #2,"25th %ile";",";
3010 print using "  #.###  ";est(pin25),adj1(pin25),adj2(pin25),adj3(pin25),adj4(pin25)
3011 PRINT #2, USING "#.###";EST(pin25);:PRINT #2,",";
3012 PRINT #2, USING "#.###";adj1(pin25);:PRINT #2,",";
3020 PRINT #2, USING "#.###";adj2(pin25);:PRINT #2,",";
3030 PRINT #2, USING "#.###";ADJ3(pin25);:PRINT #2,",";
3040 PRINT #2, USING "#.###";ADJ4(pin25)
3103 PRINT "50th %ile",
3106 print #2,"50th %ile";",";
3110 print using "  #.###  ";est(pin50),adj1(pin50),adj2(pin50),adj3(pin50),adj4(pin50)
3111 PRINT #2, USING "#.###";EST(pin50);:PRINT #2,",";
3112 PRINT #2, USING "#.###";adj1(pin50);:PRINT #2,",";
3120 PRINT #2, USING "#.###";adj2(pin50);:PRINT #2,",";
3130 PRINT #2, USING "#.###";ADJ3(pin50);:PRINT #2,",";
3140 PRINT #2, USING "#.###";ADJ4(pin50)
3203 PRINT "75th %ile",
3206 print #2,"75th %ile";",";
3210 print using "  #.###  ";est(pin75),adj1(pin75),adj2(pin75),adj3(pin75),adj4(pin75)
3211 PRINT #2, USING "#.###";EST(pin75);:PRINT #2,",";
3212 PRINT #2, USING "#.###";adj1(pin75);:PRINT #2,",";
3220 PRINT #2, USING "#.###";adj2(pin75);:PRINT #2,",";
3230 PRINT #2, USING "#.###";ADJ3(pin75);:PRINT #2,",";
3240 PRINT #2, USING "#.###";ADJ4(pin75)
3303 PRINT "90th %ile",
3306 print #2,"90th %ile";",";
3310 print using "  #.###  ";est(pin90),adj1(pin90),adj2(pin90),adj3(pin90),adj4(pin90)
3311 PRINT #2, USING "#.###";EST(pin90);:PRINT #2,",";
3312 PRINT #2, USING "#.###";adj1(pin90);:PRINT #2,",";
3320 PRINT #2, USING "#.###";adj2(pin90);:PRINT #2,",";
3330 PRINT #2, USING "#.###";ADJ3(pin90);:PRINT #2,",";
3340 PRINT #2, USING "#.###";ADJ4(pin90)
3403 PRINT "95th %ile",
3406 print #2,"95th %ile";",";
3410 print using "  #.###  ";est(pin95),adj1(pin95),adj2(pin95),adj3(pin95),adj4(pin95)
3411 PRINT #2, USING "#.###";EST(pin95);:PRINT #2,",";
3412 PRINT #2, USING "#.###";adj1(pin95);:PRINT #2,",";
3420 PRINT #2, USING "#.###";adj2(pin95);:PRINT #2,",";
3430 PRINT #2, USING "#.###";ADJ3(pin95);:PRINT #2,",";
3440 PRINT #2, USING "#.###";ADJ4(pin95)
3503 PRINT "99th %ile",
3506 print #2,"99th %ile";",";
3510 print using "  #.###  ";est(pin99),adj1(pin99),adj2(pin99),adj3(pin99),adj4(pin99)
3511 PRINT #2, USING "#.###";EST(pin99);:PRINT #2,",";
3512 PRINT #2, USING "#.###";adj1(pin99);:PRINT #2,",";
3520 PRINT #2, USING "#.###";adj2(pin99);:PRINT #2,",";
3530 PRINT #2, USING "#.###";ADJ3(pin99);:PRINT #2,",";
3540 PRINT #2, USING "#.###";ADJ4(pin99)
5968 'print:print
5969 print #2,"":print #2,""
5970 PRINT "True p: ";TRUEP
5980 PRINT #2,"True p: ";TRUEP
5990 'PRINT
6000 CLOSE #2
```

# Appendix B. Sample Program Using Monte Carlo Estimation

```
10 'Program to do monte carlo simulation for adjustment of p  (J. R. Lewis 7/04/00)
20 '
30 'The program expects to find the errors in rows separated by commas (cols are participants).
40 'The first column is the problem number.
50 'The last column is reserved for an impact rating (0 if no rating available).
60 'This version of the program gets results for participant subsets from size 2 to 6.
65 'This is the most flexible version of the pxcarlo.bas programs
70 '
71 'input "Filename for output: ",ot$
72 OT$="presults.csv"
73 OPEN OT$ FOR OUTPUT AS #2
74 let filenum=9
75 f$(1)="macerr10.asc":f$(2)="macerr25.asc":f$(3)="macerr50.asc":F$(4)="toyerr.asc"
76 f$(5)="macerr73.asc":f$(6)="macerr.asc":f$(7)="man21err.asc":f$(8)="virerr.asc"
77 f$(9)="manterr.asc"
78 for f=1 to filenum
79 for o=2 to 6
80 CLS
90 'INPUT "Filename for error data: ",F$(f)
130 OPEN F$(f) FOR INPUT AS #1
140 'INPUT "Number of identified problems/errors: ",E
150 INPUT #1, E
160 'INPUT "Number of participants: ",P
170 INPUT #1, P
180 'input "Size of participant subset: ",o
190 'O=2 'Do not set o any larger than 6! -- is done in a 2-6 loop for this version
200 LET C=1000
210 cls:print "Monte Carlo estimation of p"
211 DIM ERRORS(E,P), ERRCNT(E), EST(C), ADJ1(C), ADJ2(C), ADJ3(C), ADJ4(C), adj5(c)
212 dim curord(o),ordtrack(p),sumord(p)
213 print:print
214 print "Input file: ";f$(f)
215 print "Output file: ";ot$
220 print "Total participants: ";p
225 print "Total errors: ";e
226 print "Sample set size: ";o
228 print "Iterations: ";c
230 print:print
240 print "Iteration #:"
251 'Calculate indices for key percentiles
252 for x=1 to c
253 let perc=x*100/c
254 if pin1fnd=1 then goto 256
255 if perc>=1 then pin1=x:pin1fnd=1
256 if pin5fnd=1 then goto 258
257 if perc>=5 then pin5=x:pin5fnd=1
258 if pin10fnd=1 then goto 260
259 if perc>=10 then pin10=x:pin10fnd=1
260 if pin25fnd=1 then goto 262
261 if perc>=25 then pin25=x:pin25fnd=1
262 if pin50fnd=1 then goto 264
263 if perc>=50 then pin50=x:pin50fnd=1
264 if pin75fnd=1 then goto 266
265 if perc>=75 then pin75=x:pin75fnd=1
266 if pin90fnd=1 then goto 268
267 if perc>=90 then pin90=x:pin90fnd=1
268 if pin95fnd=1 then goto 270
269 if perc>=95 then pin95=x:pin95fnd=1
270 if pin99fnd=1 then goto 272
271 if perc>=99 then pin99=x:pin99fnd=1
272 next x
273 'print pin1,pin5,pin10,pin25,pin50,pin75,pin90,pin95,pin99:input junk
280 'Read in error matrix -- first column has err num, last column can hold impact rating
281 FOR X=1 TO E
290 INPUT #1, JUNK 'discard error id number
300 FOR Y=1 TO P 'get error occurrence for each participant
310 INPUT #1, ERRORS(X,Y)
320 NEXT Y
330 INPUT #1, JUNK 'discard impact rating
340 NEXT X
350 CLOSE #1
360 'PRINT "Table of Problem Occurrence"
370 FOR X=1 TO E
380 FOR Y=1 TO P
390 'PRINT ERRORS(X,Y);" ";
400 NEXT Y
410 'PRINT
420 NEXT X
430 'PRINT:PRINT
440 'Run monte carlo simulations, calculate and write data
460 'PRINT "Errdat Table":PRINT
465 'PRINT #2,"Errdat Table":PRINT #2,""
466 'Create the next random order
467 for cnt=1 to c           ' Clear these arrays
470 locate 12,15:print cnt
474 for y1=1 to p:let ordtrack(y1)=0:next y1
475 for y1=1 to o:let curord(y1)=0:next y1
476 for y1=1 to o
477 let ord=int(100*rnd(val(right$(time$,2))))
478 if ord>p or ord=0 then goto 477
479 if ordtrack(ord)=1 then goto 477
480 if ordtrack(ord)=0 then let curord(y1)=ord:let ordtrack(ord)=1
481 next y1
482 for y1=1 to p
483 'print ordtrack(y1);" ";
484 let sumord(y1)=sumord(y1)+ordtrack(y1)
485 next y1
486 'print:print
487 'for y1=1 to o:print curord(y1);" ";:next y1
488 if o>5 then let u=curord(6)
489 if o>4 then let v=curord(5)
490 if o>3 then let w=curord(4)
491 if o>2 then let x=curord(3)
```

```
492 let y=curord(2)
493 let z=curord(1)
494 'Clear errcnt vector
520 FOR Q=1 TO E
530 LET ERRCNT(Q)=0
540 NEXT Q
520 FOR Q=1 TO E
521 if o>5 then let errcnt(q)=errcnt(q)+errors(q,u)
522 if o>4 then let errcnt(q)=errcnt(q)+errors(q,v)
523 if o>3 then let errcnt(q)=errcnt(q)+errors(q,w)
524 if o>2 then let errcnt(q)=errcnt(q)+errors(q,x)
530 LET ERRCNT(Q)=errcnt(q)+ERRORS(Q,Y)+ERRORS(Q,Z)
540 NEXT Q
550 'PRINT u;" ";v;" ";w;" ";X;" ";Y;" ";Z;" ":FOR Q=1 TO e:PRINT ERRCNT(Q);" ";:NEXT Q
560 'Find total number of problems for this subset
570 LET PZERO=0 'initialize pzero -- the number of problem counts equal to 0
580 LET PONE=0 'initialize pone -- the number of problem counts equal to 1
590 FOR Q=1 TO E
600 IF ERRCNT(Q)=0 THEN LET PZERO=PZERO+1
610 IF ERRCNT(Q)=1 THEN LET PONE=PONE+1
620 NEXT Q
630 LET TPROB=E-PZERO  'tprob is total number of discovered problems for this subset
640 'Find total number of problem occurrences
650 LET TOCC=0 'initialize tocc -- total number of problem occurrences for this subset
660 FOR Q=1 TO E
670 LET TOCC=TOCC+ERRCNT(Q)
680 NEXT Q
690 LET TOPP=TPROB*O 'Total number of opportunities for error
700 LET ESTP=TOCC/TOPP 'Estimate of p (unadjusted)
710 LET ADJOPP1=TOPP+PONE 'Adjust opportunities with pone
720 LET ADJP1=TOCC/ADJOPP1 'Adjusted estimate of p with pone
730 LET ADJOPP2=TOPP+TPROB 'Adjust opportunities with tprob
740 LET ADJP2=TOCC/ADJOPP2 'Adjusted estimate of p with tocc
750 LET ADJOPP3=TOPP+TOCC 'Adjust opportunities with tocc
760 LET ADJP3=TOCC/ADJOPP3 'Adjusted estimate of p with tocc
770 LET ADJP4=ESTP*ADJP2+(1-ESTP)*ADJP3 'Linear interp of 2 and 3 using estp for weight
771 let goodtur=pone/tprob 'Good-Turing adjustment factor
772 let adjp5=estp/(goodtur+1) 'Make Good-Turing adjustment of p
780 'print cnt;" ";
781 'print #2, cnt;",";
788 'if o>5 then print #2, u;",";
789 'if o>4 then print #2, v;",";
790 'if o>3 then print #2, w;",";
791 'if o>2 then print #2, x;",";
792 'print #2, y;",";
793 'print #2, z;",";
800 'FOR Q=1 TO E  'Enable this loop to print out the detailed info for each iteration
810 'PRINT ERRCNT(Q);" ";
820 'PRINT #2,ERRCNT(Q);",";
830 'NEXT Q
840 'PRINT TPROB;" ";TOCC;" ";TOPP;" ";ESTP;" ";ADJP1;" ";ADJP2;" ";ADJP3;" ";ADJP4" ";ADJP5
850 'PRINT #2, TPROB;",";TOCC;",";TOPP;",";ESTP;",";ADJP1;",";ADJP2;",";ADJP3;",";ADJP4;",";ADJP5
860 LET SUMESTP=SUMESTP+ESTP:LET SUMADJP1=SUMADJP1+ADJP1
870 LET SUMADJP2=SUMADJP2+ADJP2:LET SUMADJP3=SUMADJP3+ADJP3:LET SUMADJP4=SUMADJP4+ADJP4
875 LET SUMADJP5=SUMADJP5+ADJP5
880 LET EST(CNT)=ESTP:LET ADJ1(CNT)=ADJP1:LET ADJ2(CNT)=ADJP2
882 LET ADJ3(CNT)=ADJP3:LET ADJ4(CNT)=ADJP4:let adj5(cnt)=adjp5
885 for q=1 to e:let errcnt(q)=0:next q  'clear errcnt vector
891 next cnt
892 let cnt=cnt-1
893 print #2,"":print #2,"Total Participant Selections for ";f$(f);" with";o;"samples"
894 print #2,"":print #2,"#Errors:";",";e;",";" ";",";"#Parts:";",";p
895 for y1=1 to p
896 print #2, y1;",";sumord(y1);",";c*o/p
897 let chipoint=((sumord(y1)-c*o/p)^2)/(c*o/p)
898 let chisq=chisq+chipoint
900 next y1
901 let df=p-1
910 print #2,"Chisq:";',';chisq;",";"df:";",";df;",";"Prob:" 'Use chiprob program to get prob
929 'Shell sort the p arrays
930 PRINT:PRINT "Sorting est"
931 let n=c
932 'for x=1 to n:print est(x);" ";:next x:print "906-est-unsorted":input junk
933 M8=n
934 M8=INT(M8/2)
940 IF M8=0 THEN 1040
950 K8=N-M8 : J8=1
960 I8=J8
970 L8=I8+M8
980 IF est(I8)<=est(L8) THEN 1020
990 T8=est(I8) : est(I8)=est(L8)
1000 est(L8)=T8 : I8=I8-M8
1010 IF I8>=1 THEN 970
1020 J8=J8+1
1030 IF J8<=K8 THEN 960 ELSE 934
1040 'for x=1 to n:print est(x);" ";:next x:print "1040-est-sorted":input junk
1071 PRINT "Sorting adj1"
1110 'for x=1 to n:print adj1(x);" ";:next x:print "1110-adj1-unsorted":input junk
1120 M8=n
1130 M8=INT(M8/2)
1140 IF M8=0 THEN 1240
1150 K8=N-M8 : J8=1
1160 I8=J8
1170 L8=I8+M8
1180 IF adj1(I8)<=adj1(L8) THEN 1220
1190 T8=adj1(I8) : adj1(I8)=adj1(L8)
1200 adj1(L8)=T8 : I8=I8-M8
1210 IF I8>=1 THEN 1170
1220 J8=J8+1
1230 IF J8<=K8 THEN 1160 ELSE 1130
1240 'for x=1 to n:print adj1(x);" ";:next x:print "1240-adj1-sorted":input junk
1241 PRINT "Sorting adj2"
1242 'for x=1 to c:print adj2(x);" ";:next x:print "1242-adj2-unsorted":input junk
1243 n=0 'checking that changing n to c will fix sort
1245 M8=C
1246 M8=INT(M8/2)
1247 IF M8=0 THEN goto 1330
1248 K8=c-M8 : J8=1
```

```
1250 I8=J8
1260 L8=I8+M8
1270 IF ADJ2(I8)<=ADJ2(L8) THEN 1310
1280 T8=ADJ2(I8) : ADJ2(I8)=ADJ2(L8)
1290 ADJ2(L8)=T8 : I8=I8-M8
1300 IF I8>=1 THEN 1260
1310 J8=J8+1
1320 IF J8<=K8 THEN 1250 ELSE 1246
1330 'for x=1 to c:print adj2(x);" ";:next x:print "1330-adj2-sorted":input junk
1331 PRINT "Sorting adj3"
1333 'for x=1 to c:print adj3(x);" ";:next x:print "1333-adj3-unsorted":input junk
1340 M8=C
1350 M8=INT(M8/2)
1360 IF M8=0 THEN goto 1460
1370 K8=c-M8 : J8=1
1380 I8=J8
1390 L8=I8+M8
1400 IF ADJ3(I8)<=ADJ3(L8) THEN 1440
1410 T8=ADJ3(I8) : ADJ3(I8)=ADJ3(L8)
1420 ADJ3(L8)=T8 : I8=I8-M8
1430 IF I8>=1 THEN 1390
1440 J8=J8+1
1450 IF J8<=K8 THEN 1380 ELSE 1350
1460 'for x=1 to c:print adj3(x);" ";:next x:print "1460-adj3-sorted":input junk
1461 PRINT "Sorting adj4"
1462 'for x=1 to c:print adj4(x);" ";:next x:print "1462-adj4-unsorted":input junk
1470 M8=C
1480 M8=INT(M8/2)
1490 IF M8=0 THEN goto 1581
1500 K8=c-M8 : J8=1
1510 I8=J8
1520 L8=I8+M8
1530 IF ADJ4(I8)<=ADJ4(L8) THEN 1570
1540 T8=ADJ4(I8) : ADJ4(I8)=ADJ4(L8)
1550 ADJ4(L8)=T8 : I8=I8-M8
1560 IF I8>=1 THEN 1520
1570 J8=J8+1
1580 IF J8<=K8 THEN 1510 ELSE 1480
1581 PRINT "Sorting adj5"
1582 'for x=1 to c:print adj5(x);" ";:next x:print "1462-adj5-unsorted":input junk
1583 M8=C
1584 M8=INT(M8/2)
1585 IF M8=0 THEN goto 1595
1586 K8=c-M8 : J8=1
1587 I8=J8
1588 L8=I8+M8
1589 IF ADJ5(I8)<=ADJ5(L8) THEN 1593
1590 T8=ADJ5(I8) : ADJ5(I8)=ADJ5(L8)
1591 ADJ5(L8)=T8 : I8=I8-M8
1592 IF I8>=1 THEN 1588
1593 J8=J8+1
1594 IF J8<=K8 THEN 1587 ELSE 1584
1595 'for x=1 to c:print adj4(x);" ";:next x:print "1590-adj4-sorted":input junk
1598 'Calculate final statistics
1600 FOR X=1 TO E
1610 FOR Y=1 TO P
1620 LET GRANDERR=GRANDERR+ERRORS(X,Y)
1630 LET GRANDOPP=GRANDOPP+1
1640 NEXT Y
1650 NEXT X
1660 LET TRUEP=GRANDERR/GRANDOPP
1670 LET ESTP=SUMESTP/C:LET ADJP1=SUMADJP1/C
1680 LET ADJP2=SUMADJP2/C:LET ADJP3=SUMADJP3/C:LET ADJP4=SUMADJP4/C:LET ADJP5=SUMADJP5/C
1690 FOR X=1 TO C
1700 LET VARESTP=VARESTP+(EST(X)-ESTP)^2
1710 LET VARADJP1=VARADJP1+(ADJ1(X)-ADJP1)^2
1720 LET VARADJP2=VARADJP2+(ADJ2(X)-ADJP2)^2
1730 LET VARADJP3=VARADJP3+(ADJ3(X)-ADJP3)^2
1740 LET VARADJP4=VARADJP4+(ADJ4(X)-ADJP4)^2
1744 let VARADJP5=VARADJP5+(ADJ5(X)-ADJP5)^2
1745 LET RMSestp=RMSestp+(est(X)-truep)^2
1750 LET RMSADJP1=RMSADJP1+(ADJ1(X)-truep)^2
1760 LET RMSADJP2=RMSADJP2+(ADJ2(X)-truep)^2
1770 LET RMSADJP3=RMSADJP3+(ADJ3(X)-truep)^2
1780 LET RMSADJP4=RMSADJP4+(ADJ4(X)-truep)^2
1785 let RMSADJP5=RMSADJP5+(ADJ5(X)-truep)^2
1790 NEXT X
1800 LET VARESTP=VARESTP/(C-1):LET VARADJP1=VARADJP1/(C-1)
1810 LET VARADJP2=VARADJP2/(C-1):LET VARADJP3=VARADJP3/(C-1):LET VARADJP4=VARADJP4/(C-1)
1815 LET VARADJP5=VARADJP5/(C-1)
1820 LET SDESTP=VARESTP^.5:LET SDADJP1=VARADJP1^.5
1830 LET SDADJP2=VARADJP2^.5:LET SDADJP3=VARADJP3^.5:LET SDADJP4=VARADJP4^.5
1835 let SDADJP5=VARADJP5^.5
1840 LET SEMESTP=SDESTP/(C^.5):LET SEMADJP1=SDADJP1/(C^.5)
1850 LET SEMADJP2=SDADJP2/(C^.5):LET SEMADJP3=SDADJP3/(C^.5):LET SEMADJP4=SDADJP4/(C^.5)
1855 let SEMADJP5=SDADJP5/(C^.5)
1860 LET Z99=2.576
1870 LET D99ESTP=Z99*SEMESTP:LET D99ADJP1=Z99*SEMADJP1
1880 LET D99ADJP2=Z99*SEMADJP2:LET D99ADJP3=Z99*SEMADJP3:LET D99ADJP4=Z99*SEMADJP4
1885 let D99ADJP5=Z99*SEMADJP5
1890 LET UP99EST=ESTP+D99ESTP:LET UP99ADJ1=ADJP1+D99ADJP1
1900 LET UP99ADJ2=ADJP2+D99ADJP2:LET UP99ADJ3=ADJP3+D99ADJP3:LET UP99ADJ4=ADJP4+D99ADJP4
1905 let UP99ADJ5=ADJP5+D99ADJP5
1910 LET LOW99EST=ESTP-D99ESTP:LET LOW99ADJ1=ADJP1-D99ADJP1
1920 LET LOW99ADJ2=ADJP2-D99ADJP2:LET LOW99ADJ3=ADJP3-D99ADJP3:LET LOW99ADJ4=ADJP4-D99ADJP4
1925 let LOW99ADJ5=ADJP5-D99ADJP5
1930 'Print stats
1940 CLS
1950 PRINT "Statistic ","Original ";"Adj-Ones ";"Adj-Prbs ";"Adj-Occs ";"Adj-LinOp";" Adj-GT"
1960 PRINT #2,"Statistic";",";"Original";",";"Adj-Ones";",";"Adj-Probs";",";"Adj-Occs";",";
1970 PRINT #2,"Adj-LinOp";",";"Adj-GT"
1980 PRINT "--------- ","-------- ";"-------- ";"-------- ";"-------- ";"---------";" --------"
1990 PRINT "Mean",
2000 PRINT #2,"Mean";",";
2010 PRINT USING "  #.###  ";ESTP,ADJP1,ADJP2,ADJP3,ADJP4,adjp5
2020 PRINT #2, USING "#.###";ESTP;:PRINT #2,",";
2030 PRINT #2, USING "#.###";ADJP1;:PRINT #2,",";
2040 PRINT #2, USING "#.###";ADJP2;:PRINT #2,",";
```

```
2050 PRINT #2, USING "#.###";ADJP3;:PRINT #2,",";
2060 PRINT #2, USING "#.###";ADJP4;:PRINT #2,",";
2065 print #2, using "#.###";adjp5
2070 PRINT "Std Dev",
2080 PRINT #2,"Std Dev";",";
2090 PRINT USING "  #.###  ";SDESTP,SDADJP1,SDADJP2,SDADJP3,SDADJP4,sdadjp5
2100 PRINT #2, USING "#.###";SDESTP;:PRINT #2,",";
2110 PRINT #2, USING "#.###";SDADJP1;:PRINT #2,",";
2120 PRINT #2, USING "#.###";SDADJP2;:PRINT #2,",";
2130 PRINT #2, USING "#.###";SDADJP3;:PRINT #2,",";
2135 PRINT #2, USING "#.###";SDADJP4;:PRINT #2,",";
2140 PRINT #2, USING "#.###";SDADJP5
2150 PRINT "RMS Err",
2160 PRINT #2,"RMS Err";",";
2161 let rmsadjp1=((rmsadjp1/(c-1))^.5)
2162 let rmsadjp2=((rmsadjp2/(c-1))^.5)
2163 let rmsadjp3=((rmsadjp3/(c-1))^.5)
2164 let rmsadjp4=((rmsadjp4/(c-1))^.5)
2165 let rmsadjp5=((rmsadjp5/(c-1))^.5)
2166 let rmsestp=((rmsestp/(c-1))^.5)
2170 PRINT USING "  #.###  ";rmsESTP,RMSADJP1,RMSADJP2,RMSADJP3,RMSADJP4,RMSADJP5
2180 PRINT #2, USING "#.###";rmsESTP;:PRINT #2,",";
2190 PRINT #2, USING "#.###";RMSADJP1;:PRINT #2,",";
2200 PRINT #2, USING "#.###";RMSADJP2;:PRINT #2,",";
2210 PRINT #2, USING "#.###";RMSADJP3;:PRINT #2,",";
2215 PRINT #2, USING "#.###";RMSADJP4;:PRINT #2,",";
2220 PRINT #2, USING "#.###";RMSADJP5
2230 PRINT "sem",
2240 PRINT #2, "sem";",";
2250 PRINT USING "  #.###  ";SEMESTP,SEMADJP1,SEMADJP2,SEMADJP3,SEMADJP4,semadjp5
2260 PRINT #2, USING "#.###";SEMESTP;:PRINT #2,",";
2270 PRINT #2, USING "#.###";SEMADJP1;:PRINT #2,",";
2280 PRINT #2, USING "#.###";SEMADJP2;:PRINT #2,",";
2290 PRINT #2, USING "#.###";SEMADJP3;:PRINT #2,",";
2295 PRINT #2, USING "#.###";SEMADJP4;:PRINT #2,",";
2300 PRINT #2, USING "#.###";SEMADJP5
2310 PRINT "d99",
2320 PRINT #2, "d99";",";
2330 PRINT USING "  #.###  ";D99ESTP,D99ADJP1,D99ADJP2,D99ADJP3,D99ADJP4,d99adjp5
2340 PRINT #2, USING "#.###";D99ESTP;:PRINT #2,",";
2350 PRINT #2, USING "#.###";D99ADJP1;:PRINT #2,",";
2360 PRINT #2, USING "#.###";D99ADJP2;:PRINT #2,",";
2370 PRINT #2, USING "#.###";D99ADJP3;:PRINT #2,",";
2375 PRINT #2, USING "#.###";D99ADJP4;:PRINT #2,",";
2380 PRINT #2, USING "#.###";D99ADJP5
2390 'PRINT
2400 PRINT #2,""
2410 PRINT "Upper",
2420 PRINT #2, "Upper";",";
2430 PRINT USING "  #.###  ";UP99EST,UP99ADJ1,UP99ADJ2,UP99ADJ3,UP99ADJ4,UP99ADJ5
2440 PRINT #2, USING "#.###";UP99EST;:PRINT #2,",";
2450 PRINT #2, USING "#.###";UP99ADJ1;:PRINT #2,",";
2460 PRINT #2, USING "#.###";UP99ADJ2;:PRINT #2,",";
2470 PRINT #2, USING "#.###";UP99ADJ3;:PRINT #2,",";
2475 PRINT #2, USING "#.###";UP99ADJ4;:PRINT #2,",";
2480 PRINT #2, USING "#.###";UP99ADJ5
2490 PRINT "Mean",
2500 PRINT #2, "Mean";",";
2510 PRINT USING "  #.###  ";ESTP,ADJP1,ADJP2,ADJP3,ADJP4,adjp5
2520 PRINT #2, USING "#.###";ESTP;:PRINT #2,",";
2530 PRINT #2, USING "#.###";ADJP1;:PRINT #2,",";
2540 PRINT #2, USING "#.###";ADJP2;:PRINT #2,",";
2550 PRINT #2, USING "#.###";ADJP3;:PRINT #2,",";
2555 PRINT #2, USING "#.###";ADJP4;:PRINT #2,",";
2560 PRINT #2, USING "#.###";ADJP5
2570 PRINT "Lower",
2580 PRINT #2, "Lower";",";
2590 PRINT USING "  #.###  ";LOW99EST,LOW99ADJ1,LOW99ADJ2,LOW99ADJ3,LOW99ADJ4,LOW99ADJ5
2600 PRINT #2, USING "#.###";LOW99EST;:PRINT #2,",";
2610 PRINT #2, USING "#.###";LOW99ADJ1;:PRINT #2,",";
2620 PRINT #2, USING "#.###";LOW99ADJ2;:PRINT #2,",";
2630 PRINT #2, USING "#.###";LOW99ADJ3;:PRINT #2,",";
2635 PRINT #2, USING "#.###";LOW99ADJ4;:PRINT #2,",";
2640 PRINT #2, USING "#.###";LOW99ADJ5
2650 PRINT:PRINT
2660 PRINT #2, ""
2700 'print percentiles
2703 PRINT "1st %ile",
2706 print #2,"1st %ile";",";
2710 print using "  #.###  ";est(pin1),adj1(pin1),adj2(pin1),adj3(pin1),adj4(pin1),adj5(pin1)
2711 PRINT #2, USING "#.###";EST(pin1);:PRINT #2,",";
2712 PRINT #2, USING "#.###";adj1(pin1);:PRINT #2,",";
2720 PRINT #2, USING "#.###";adj2(pin1);:PRINT #2,",";
2730 PRINT #2, USING "#.###";ADJ3(pin1);:PRINT #2,",";
2735 PRINT #2, USING "#.###";ADJ4(pin1);:PRINT #2,",";
2740 PRINT #2, USING "#.###";ADJ5(pin1)
2803 PRINT "5th %ile",
2806 print #2,"5th %ile";",";
2810 print using "  #.###  ";est(pin5),adj1(pin5),adj2(pin5),adj3(pin5),adj4(pin5),adj5(pin5)
2811 PRINT #2, USING "#.###";EST(pin5);:PRINT #2,",";
2812 PRINT #2, USING "#.###";adj1(pin5);:PRINT #2,",";
2820 PRINT #2, USING "#.###";adj2(pin5);:PRINT #2,",";
2830 PRINT #2, USING "#.###";ADJ3(pin5);:PRINT #2,",";
2835 PRINT #2, USING "#.###";ADJ4(pin5);:PRINT #2,",";
2840 PRINT #2, USING "#.###";ADJ5(pin5)
2903 PRINT "10th %ile",
2906 print #2,"10th %ile";",";
2910 print using "  #.###  ";est(pin10),adj1(pin10),adj2(pin10),adj3(pin10),adj4(pin10),adj5(pin10)
2911 PRINT #2, USING "#.###";EST(pin10);:PRINT #2,",";
2912 PRINT #2, USING "#.###";adj1(pin10);:PRINT #2,",";
2920 PRINT #2, USING "#.###";adj2(pin10);:PRINT #2,",";
2930 PRINT #2, USING "#.###";ADJ3(pin10);:PRINT #2,",";
2935 PRINT #2, USING "#.###";ADJ4(pin10);:PRINT #2,",";
2940 PRINT #2, USING "#.###";ADJ5(pin10)
3003 PRINT "25th %ile",
3006 print #2,"25th %ile";",";
3010 print using "  #.###  ";est(pin25),adj1(pin25),adj2(pin25),adj3(pin25),adj4(pin25),adj5(pin25)
3011 PRINT #2, USING "#.###";EST(pin25);:PRINT #2,",";
```

```
3012 PRINT #2, USING "#.###";adj1(pin25);:PRINT #2,",";
3020 PRINT #2, USING "#.###";adj2(pin25);:PRINT #2,",";
3030 PRINT #2, USING "#.###";ADJ3(pin25);:PRINT #2,",";
3035 PRINT #2, USING "#.###";ADJ4(pin25);:PRINT #2,",";
3040 PRINT #2, USING "#.###";ADJ5(pin25)
3103 PRINT "50th %ile",
3106 print #2,"50th %ile";",";
3110 print using "  #.###  ";est(pin50),adj1(pin50),adj2(pin50),adj3(pin50),adj4(pin50),adj5(pin50)
3111 PRINT #2, USING "#.###";EST(pin50);:PRINT #2,",";
3112 PRINT #2, USING "#.###";adj1(pin50);:PRINT #2,",";
3120 PRINT #2, USING "#.###";adj2(pin50);:PRINT #2,",";
3130 PRINT #2, USING "#.###";ADJ3(pin50);:PRINT #2,",";
3135 PRINT #2, USING "#.###";ADJ4(pin50);:PRINT #2,",";
3140 PRINT #2, USING "#.###";ADJ5(pin50)
3203 PRINT "75th %ile",
3206 print #2,"75th %ile";",";
3210 print using "  #.###  ";est(pin75),adj1(pin75),adj2(pin75),adj3(pin75),adj4(pin75),adj5(pin75)
3211 PRINT #2, USING "#.###";EST(pin75);:PRINT #2,",";
3212 PRINT #2, USING "#.###";adj1(pin75);:PRINT #2,",";
3220 PRINT #2, USING "#.###";adj2(pin75);:PRINT #2,",";
3230 PRINT #2, USING "#.###";ADJ3(pin75);:PRINT #2,",";
3235 PRINT #2, USING "#.###";ADJ4(pin75);:PRINT #2,",";
3240 PRINT #2, USING "#.###";ADJ5(pin75)
3303 PRINT "90th %ile",
3306 print #2,"90th %ile";",";
3310 print using "  #.###  ";est(pin90),adj1(pin90),adj2(pin90),adj3(pin90),adj4(pin90),adj5(pin90)
3311 PRINT #2, USING "#.###";EST(pin90);:PRINT #2,",";
3312 PRINT #2, USING "#.###";adj1(pin90);:PRINT #2,",";
3320 PRINT #2, USING "#.###";adj2(pin90);:PRINT #2,",";
3330 PRINT #2, USING "#.###";ADJ3(pin90);:PRINT #2,",";
3335 PRINT #2, USING "#.###";ADJ4(pin90);:PRINT #2,",";
3340 PRINT #2, USING "#.###";ADJ5(pin90)
3403 PRINT "95th %ile",
3406 print #2,"95th %ile";",";
3410 print using "  #.###  ";est(pin95),adj1(pin95),adj2(pin95),adj3(pin95),adj4(pin95),adj5(pin95)
3411 PRINT #2, USING "#.###";EST(pin95);:PRINT #2,",";
3412 PRINT #2, USING "#.###";adj1(pin95);:PRINT #2,",";
3420 PRINT #2, USING "#.###";adj2(pin95);:PRINT #2,",";
3430 PRINT #2, USING "#.###";ADJ3(pin95);:PRINT #2,",";
3435 PRINT #2, USING "#.###";ADJ4(pin95);:PRINT #2,",";
3440 PRINT #2, USING "#.###";ADJ5(pin95)
3503 PRINT "99th %ile",
3506 print #2,"99th %ile";",";
3510 print using "  #.###  "; est(pin99), adj1(pin99), adj2(pin99), adj3(pin99), adj4(pin99), adj5(pin99)
3511 PRINT #2, USING "#.###";EST(pin99);:PRINT #2,",";
3512 PRINT #2, USING "#.###";adj1(pin99);:PRINT #2,",";
3520 PRINT #2, USING "#.###";adj2(pin99);:PRINT #2,",";
3530 PRINT #2, USING "#.###";ADJ3(pin99);:PRINT #2,",";
3535 PRINT #2, USING "#.###";ADJ4(pin99);:PRINT #2,",";
3540 PRINT #2, USING "#.###";ADJ5(pin99)
5968 'print:print
5969 print #2,""
5970 PRINT "File : ";f$(f);"   Sample size (o): ";o;"   True p: ";TRUEP
5981 print #2,"File: ";",";f$(f);",";"Sample size (o): ";",";o;",";"Runs (c): ";",";c;
5982 PRINT #2,"True p: ";",";TRUEP;","
5983 print #2,"":print #2,"":print #2,""
5990 'PRINT
5992 erase ERRORS:erase ERRCNT:erase EST:erase ADJ1:erase ADJ2:erase ADJ3:erase ADJ4
5993 erase adj5:erase curord:erase ordtrack:erase sumord
5994 let sumestp=0:let sumadjp1=0:let sumadjp2=0:let sumadjp3=0:let sumadjp4=0:let sumadjp5=0
5995 let varestp=0:let varadjp1=0:let varadjp2=0:let varadjp3=0:let varadjp4=0:let varadjp5=0
5996 let rmsestp=0:let rmsadjp1=0:let rmsadjp2=0:let rmsadjp3=0:let rmsadjp4=0:let rmsadjp5=0
5997 let chisq=0:let granderr=0:let grandopp=0
5999 next o
6000 next f
6010 close #2
```

## Appendix C. The Problem-Discovery Databases

In every database, the first two values are (1) the number of problems and (2) the number of participants in the database. After those values, the leftmost column is a list of problem identification numbers and the rightmost column indicates problem impact (severity), if available. If impact ratings were not available, the rightmost column contains all zeros. The intervening columns represent the participants in the study. In the cells of the participant columns, a ‘1’ indicates that the participant experienced the identified problem during the usability evaluation and a ‘0’ indicates that the participant did not experience that problem.

**SAMPLE**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 10 | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**MACERR**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 145 | 15 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 2 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 2 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 4 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 4 |
| 5 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 3 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 |
| 7 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 3 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 14 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| 17 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 18 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 2 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 21 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 22 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 23 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 27 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| 29 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 3 |
| 30 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 31 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 32 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 |
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3 |
| 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| 35 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 37 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| 38 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 |
| 39 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 41 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 42 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 43 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 44 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 46 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 47 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 49 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 50 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 51 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 52 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 54 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| 56 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 2 |
| 57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| 58 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 59 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 60 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 61 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 62 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 63 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 64 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 65 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 66 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 67 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 68 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 69 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 |
| 70 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 71 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 72 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 73 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 74 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 75 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| 76 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 77 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 78 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 79 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 80 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 81 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 82 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 83 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 84 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 85 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 86 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 87 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 88 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 89 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 90 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 91 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 92 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 93 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 94 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 95 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 96 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 97 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 102 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 103 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 104 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 105 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 106 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 107 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 108 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 109 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 110 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 111 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 112 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 113 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 114 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 115 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 116 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 117 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 118 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 119 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 120 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 121 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 122 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 123 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 124 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 125 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 126 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 127 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 128 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 129 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 130 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 131 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 132 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 133 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 134 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 135 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 136 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 137 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 138 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 139 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 140 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 141 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 142 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 143 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 144 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 145 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**MACERR10**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 45 | 15 | | | | | | | | | | | | | | | |
| 116 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 117 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 122 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 123 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 125 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 126 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 127 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 128 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 129 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 130 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 131 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 133 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 134 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 136 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 137 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 138 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 139 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 140 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 141 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 142 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 143 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 144 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 145 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 14 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 22 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 3 |
| 39 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 41 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 43 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 52 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 60 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 72 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 73 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 74 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 77 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 80 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 81 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 88 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 109 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 114 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 118 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 120 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 135 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

## MACERR25

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 15 | | | | | | | | | | | | | | | |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 21 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 32 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 |
| 33 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3 |
| 42 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 46 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 49 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 66 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 70 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 75 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| 93 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 94 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| 97 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 110 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 121 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 124 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 132 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 35 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 38 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 |
| 65 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 67 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 68 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 82 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 85 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 89 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 102 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 113 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 119 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 54 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| 78 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 91 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 31 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 37 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4 |

**MACERR50**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 15 | | | | | | | | | | | | | | | |
| 5 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 3 |
| 7 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 3 |
| 51 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 56 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 2 |
| 92 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| 104 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 107 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 2 |
| 29 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 3 |
| 4 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 4 |

**MACERR73**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 15 | | | | | | | | | | | | | | | |
| 27 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 2 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 2 |
| 18 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 2 |

**VIRZI90**

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 40 | 20 | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 3 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 7 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 9 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 12 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 13 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 14 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 15 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 19 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 22 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 23 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 24 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 25 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 27 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 28 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 30 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 32 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 33 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 34 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 36 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

## MANTEL

| 30 | 76 | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 7 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 9 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 11 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |