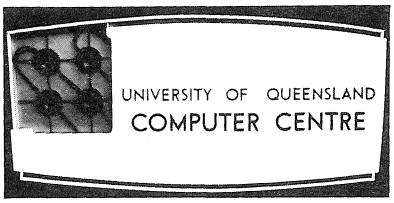

COMPUTER CENTRE BULLETIN

Volume 3, Number 4.
6th April, 1970.

THIS EDITION

Last month, Mrs. H.L. Smythe, Editor of the Bulletin, resigned from the Computer Centre to begin a school teaching career. Helen's leaving will be a loss to the Centre and, until a new Editor is appointed, the scope of the Bulletin, which Helen had built up over the past twelve months or so, will be somewhat restricted.

This issue includes the announcement of new facilities for PDP 10 users, a list of Computer Science books recently acquired by the Library, the text of a recent seminar and another *Bulletin Baffler*.

LETTER TO THE EDITOR

In the October edition of the "Computer Centre Bulletin" (6th October, 1969) in a reply to a letter about the poor data preparation facilities at the Centre it was stated that "the Centre is essentially a scientific data processing centre, not a data preparation centre".

The mind boggles!!

Any resemblance between the Computer Centre and a data processing centre is purely coincidental. For a person working a normal day the present arrangement, whereby there is a batch run on the PDP 10 at 9.00A.M. and "approximately" 4.30P.M., means an average of one run a day. While this causes considerable delay and inconvenience to the user, especially post-graduates who have scholarships for a limited time only, it is also an inefficient use of the capital tied up in hardware.

What with increased computing charges, an estimated minimum of 50% of the week during which the computer is not used, and "a token" data preparation service, surely it is not too much to expect a more regular service (?) than at present. *John R. Woodhead, P/G Student, Chemical Engineering Department, University of Queensland.*

EDITOR'S REPLY

Firstly, let me say that your letter and request represents the feelings of many users of the Centre, and Computer Centre staff are aware that the service has serious limitations.

Our log shows that for the six-week period preceding the date of your letter, there had been an average of 3.6 batch runs per day. The Centre has deliberately never officially announced batch times or the number of runs, because long periods of time are needed for development work and it is difficult to guarantee machine availability at this stage. The delay in what had become known as the afternoon run was necessary for detailed checking of jobs run under a new monitor system, as explained in the vestibule notice.

Wherever possible, future batch operation will be regarded as a batch period in which as many jobs as possible will be accepted and run; thus, users who are prepared to remain in the Centre during a batch period may be able to rerun jobs several times.

Your statement that capital is tied up in idle equipment is of course true for practically all capital equipment when considered on a 24 hour basis. The equipment at the Centre is normally running from 7.00 a.m. to 12.00 midnight and batch operation is only one of many activities. An idle card reader or line printer or the "red-light" indicator does not mean that the installation is idle.

It would, of course, be necessary for users wishing to make use of evening batch facilities to adjust their work schedule - a normal working day to many computer users does not finish at 5.00 p.m.

STAFF OF THE COMPUTER CENTRE

This month we welcome *Bill Louvel* who has joined the Maintenance Staff from Western Australia where he was an Army technician.

NEW FACILITIES FOR PDP/10 USERS

With the development of software, a number of new options have been made available to PDP/10 users.

The new options apply to both FORTRAN compilations and the running of compiled programs, and will be available as from Monday, 13th April, 1970.

(a) FORTRAN

For any FORTRAN compilation, the type of output listing required can now be specified. The following options are available:

(i) NOLIST - The FORTRAN program will be compiled, but no listing of the source program will be produced.

(ii) MACRO - During compilation, both the FORTRAN source program and the equivalent PDP/10 assembler language code produced for each FORTRAN statement by the compiler will be listed.

(iii) CREF - This will produce a sequence numbered listing of the source program followed by tables giving cross-references to all variables, statements, numbers, routine calls, etc. This type of listing can be very useful in program debugging.

(iv) If no option is specified, the source program will be listed as at present.

The options required are specified on the .FORTRAN control card, and enclosed in brackets following the verb. Note that more than one option may be included, as shown in the second example.

EXAMPLES:

.FORTRAN (NOLIST)	will compile, but not list a program
.FORTRAN (MACRO, CREF)	will produce both a MACRO and CREF listing of the source program during compilation
.FORTRAN (MACRO)	lists the FORTRAN source program and the MACRO code produced by the FORTRAN compiler

(b) PROGRAM RUNNING

The PDP/10 loader has now been modified to produce a standard memory map on the loading of a program. Additionally, a detailed symbol list, giving the location of all local and global variables, can be invoked. This is an invaluable aid in finding errors and it is recommended that it always be produced during program debugging. If no option is specified, a minimal memory map, which simply gives the starting address of each of the routines used, will be produced. The options available are:

(i) NOMAP - No loader map will be produced.

(ii) SYMBOL - Produces the detailed symbol list which gives the locations of all symbols, variables, statements, etc. The list is produced after the program execution.

The required option is specified on the .RUN card and is enclosed in brackets following the verb.

EXAMPLES:

.RUN (NOMAP)	-	no memory map is produced
.RUN	-	the minimal map is output
.RUN (SYMBOL)	-	both the summary map and the symbol list are produced

It should be noted that in all cases of options being specified, there must be at least one blank between the last character of the verb and the opening bracket.

As further facilities become available on the PDP/10, details will be announced in the Bulletin and an announcement will also be posted on the Computer Centre notice board in the vestibule of the Computer Centre.

These facilities do involve a change to a new system and, while every effort has been made to remove errors, they, no doubt, will still occur. We would appreciate it if we could be told of any errors that are found.

LIBRARY ACCESSIONS

This month's Bulletin lists the books and periodicals that the Libraries of the University of Queensland acquired in January and February 1970.

GUE, Ronald L.	*Mathematical Methods in Operations Research.* 1968. (001.424 GUE, Maths Lib.)
ROBISON, Gerson B.	*An Introduction to Mathematical Logic.* 1969. (164 ROB, Main Library)
Computer International.	(621.38195 COM, Engin.Lib.)
Joint Computer Conference, Proceedings.	1968. (621.001517 JOI, Engin.Lib.)
The Society for Information Display, Proceedings.	1969. (621.389 NAT, Engin.Lib.)
Carlson, A. Bruce	*Communication Systems.* 1968. (621.38 CAR, Engin.Lib.)
Cole, R. Wade	*Introduction to Computing.* 1969. (651.8 COL, Engin.Lib.)
Joslin, Edward O.	*Computer Selection.* 1968. (651.8 JOS, Engin.Lib.)
Walsh, Dorothy	*A Guide for Software Documentation.* 1969. (Qto 658.505 WAL, Engin.Lib.)

Computing Surveys. 1969. (651.8 COM, Engin.Lib.)

Tokyo Daigaku, Computer Centre, Report No. 2. 1968. (651.82 TOK, Engin.Lib.)

Wagner, Harvey M. *Principles of Operations Research.* 1969. (001.424 WAG, Main Lib.)

Fiacco, Anthony V. *Nonlinear Programming.* 1968. (519.92 FIA, Elect. Engin.Lib.)

Galler, Bernard A. *The Language of Computers.* 1962. (510.783 GAL, Engin.Lib.)

Iverson, Kenneth E. *The Role of Computers in Teaching.* 1968. (Qto 510.07 IVE, Maths Lib.)

Jansson, Birger *Random Number Generators.* 1966. (519.93 JAN, Maths Lib.)

Huelsman, Lawrence P. *Digital Computations in Basic Circuit Theory.* 1968. (621.31920182 HUE, Engin.Lib.)

Iri, Masao *Network Flow, Transportation, and Scheduling.* 1969. (620 IRI, Maths Lib.)

Barron, David William *Recursive Techniques in Programming.* 1968. (651.8 BAR, Engin.Lib.)

Carsberg, Bryan V. *An Introduction to Mathematical Programming for Accountants.* (657.42018 CAR, Main Lib.)

Dippel, Gene *Information Systems.* 1969. (651.8 DIP, Main Lib.)

Levison, Michael *Introduction to Computer Science.* 1968. (651.8 LEV, Main Lib.)

Lowe, Cecil William *Critical Path Analysis by Bar Chart.* 1969. (658.4 LOW, Engin.Lib.)

Sammet, Jean E. *Programming Languages.* 1969. (651.8 SAM, Main Lib.)

Stark, Peter A. *Digital Computer Programming.* 1967. (651.8 STA, Engin.Lib.)

Wiest, Jerome D. *A Management Guide to PERT/CPM.* 1969. (658.502 WIE, Main Lib.)

Wilkes, Maurice Vincent *Time-sharing Computer Systems.* 1968. (651.8 WIL, Engin.Lib.)

BULLETIN BAFFLERS

The answer to last month's Baffler is:

2 & 6

Reason:

Variable 'a' is not a formal parameter of the procedure DOUBLA. Therefore, the 'a' in the procedure represents the 'a' which is global to the procedure, i.e., the 'a' declared in the outer block.

The procedure call causes variable 'a' in the outer block to be doubled.

The first output statement prints the value of the variable 'a' which is local to the inner block, i.e., 2.

The second output statement prints the value of variable 'a' which is local to the outer block, i.e., 6.

Now, really test your ALGOL:

```
begin integer a;

procedure INCV(x); value x; integer x;  x: = x+1;

procedure INCN(x); integer x;  x: = x+1

procedure ADD(x); integer x;  x: = x+x

a: = 1;  Output (ADD(INCV(a)));  a: = 1;  Output (ADD(INCN(a)));  end;
```

What values are output by the above code and why?

We hope another *Bulletin Baffler* will appear in the next issue, along with the answer to this month's.

PROBLEM SOLVING BY SIMULATION

This is an edited version of a seminar given by Mr. I. Oliver at the University of Queensland on 18th March, 1970. Mr. Oliver, formerly a lecturer at the Computer Centre, University of Queensland is now a computer consultant in Brisbane.

There is a suggestion that the term "simulation" is not well known so that to begin we must define the term. An explanation may be forthcoming through a series of questions and answers.

Q: What is simulation?

Simple Answer: The construction of a model.

Q: What is a model?

A: A representation of the thing which embodies in some form the properties of the thing which we are studying.

Q: What on earth does that answer mean?

An answer to the last question may be gained by looking at some of the models in use around us.

EXAMPLE: Architects' models. In what way are they related to real live buildings? The similarities are shape and colour but their dissimilarities form a very long list indeed. The model serves its purpose, however, because the similar properties are all that are of interest to the architect at that stage. Other models include globes of the world, ships in bottles, toy animals etc.

All other models are called *iconic* because they look like what they represent, i.e. the visual properties of these models are the most important for the purpose to which the models are put. We also have working models; for example, model railway train sets. The most useful working models in industry are, for example, pilot chemical plants, while in Universities the Civil Engineers construct model water supply and reticulation systems. The reasons for using models are obvious

(i) convenience

(ii) saving in cost.

What is perhaps not so obvious to the uninitiated is that all physical science proceeds by the construction of models. Many of these are not expressed in a concrete form such as a pilot chemical plant. Many models are expressed most conveniently mathematically. e.g. Einstein's famous formula $E = mc^2$ is not a statement of fact but a model which displays the relationship between mass and energy. In the so called scientific method we do the following:

(i) From observation and intuition we construct a model which attempts to explain certain phenomena. The model may be a set of mathematical equations or not but it attempts to exhibit relationships between certain physical quantities.

(ii) Design experiments which can be applied to the model. We do this to try to prove the validity of the model. We must be strictly honest with ourselves so that the outcome of the experiment will in fact provide some evidence to the validity or otherwise of the model.

(iii) The second process above never stops in science. When the model is found to be defective it is amended or replaced and we go back to square 1.

The major point here is that we never really know anything. We construct models and while these are self consistent and stand up to experiments designed to test their validity we incline to accept them as facts.

Computer Simulation models are working models of real live situations. They are different from the child's train set only that they deal generally with the numerical values of the variable in the model rather than the physical bits of equipment. We all know that a computer can step through complicated series of decisions and vary the values of certain quantities on the way through. Let us see now how this can be put to use in a simple simulation model.

Let's consider a simple problem. In a supermarket there are three check-out lanes where the girls check out items at the rate of 16, 20 and 24 items per minute, plus a roughly consistent time of half a minute for paying and giving change. Customers are not aware of check-out speed. How good is the service we can provide the customers under varying loading conditions. Firstly we have to decide what we mean by different loading conditions.

A way to obtain this information is with a stop watch. We write down the time of arrival of people at the check-out lanes in an existing supermarket and count the number of items each person has. Let us suppose we have produced such a list somewhere or other.

Person arrives at check-out	Time	with	Items
1	12 hours	30 mins. 36 secs.	27
2	12 "	30 " 39 "	12
3	12 "	30 " 43 "	22
4	12 "	30 " 51 "	18
5	12 "	31 " 20 "	17
6	12 "	31 " 20 "	5
7	12 "	31 " 21 "	18

Now we can write a computer program which initially assumes that the three lanes are empty. The program reads the above data from punched cards and on reading the first card it assigns the first person to the first lane and notes that the person will leave the lane at 12 hours 30 mins. 36 secs. + $\frac{27}{16}$ + ½ min.

The program reads the second card assigns the second person to the second lane and notes the time that that person will leave. Similarly the third person is assigned to the third lane as he arrives. The program accepts the fourth person at 12 hours 30 mins. 51 secs. At this time the lane status is

Lane	Time	Items left to be checked
1	15 secs. since person entered	23
2	12 "	8
3	8 "	19

The program will therefore assign the fourth person to the second lane and put him in a queue there. This process proceeds, the computer keeping track of the times people enter and leave the lanes and assigning people entering the lanes to the line with the shortest queue or if there is no queue to the line with the fewest items left to be checked.

In this way the computer has *simulated* the check-out system of a supermarket. The assumptions in this model have already been discussed.

Clearly it is a crude model. We have not taken into account such things as tea-breaks, girls not knowing the prices of various items and the fact that we are working entirely on an average check-out rate. True check-out rates may fluctuate throughout the day.

However, the model can be made as simple or as sophisticated as one desires depending on the purpose. All that is usually required is some acceptable level of accuracy.

Although we have constructed this computer simulation model we have not made *use* of it. To make use of it we need the computer to collect all the statistics as it goes through the simulation process. For example, it can easily record

(1) the maximum queue lengths at each lane,

(2) the idle periods for each lane and

(3) the elapsed time for which there was a queue of 0,1,2,3... people.

This kind of information is the reason we carry out simulation at all.

Getting back to Science for a moment, simulation provides the most powerful means of testing hypotheses in the non-physical sciences. Consider for example, Economics. One can theorize about what happens when interest rates are raised or the Statutary Reserve Deposit increased, but there is so little scope to test such theories. By creating simulation models of the aspects of the economy under consideration some means of experimentation is provided. The ability to experiment with simulation models is in many cases the major justification.

To come back to the supermarket example. Suppose a company plans to build a supermarket and has to decide how many lanes to build. It may have various estimates of the loading from other supermarkets surveys. A computer simulation model can be built which allows runs with different numbers of lanes, different check rates, (e.g., it may be worthwhile to pay more for faster girls) and different loadings. The program can be fed with data on cost of lane construction, wages etc. and can produce a financial analysis of the possibilities in a form readily digestible by management.

In many cases if there is a large number of variables which are to be considered a competent statistician should be engaged to design an experiment so that the maximum information can be extracted from the minimum number of computer runs.

In other cases with a large number of variables it is best to use what have been called evolutionary techniques. This is where the user sets the values of all the variables and carries out simulation runs. He then carefully examines the output and on the basis of these runs may make changes to some of the variables in a systematic manner (i.e. under the guidance of a statistician) and have a number of further runs. Based on the runs which show improved performance, he makes further changes and so on. In this way the variables in a model *evolve* towards an optimal level.