COMPUTER EDUCATION.

Bulletin of the
Computer Education Group
of the
British Computer Society.

September, 1967.

00101

LIST OF CONTENTS

Title	Page
Editor's Notes	2
Committee and Constitution	3
Aims of Computer Education Group	6
Letters to the Editor	8
Conferences and Courses	15
B.C.S. Working Party No. 4	19
Education of Scientific Programmers	20
Computing Centre at Enfield College	23
Jack-of-All-Trades (J. Bradshaw)	27
Computer Education (Miss D. Law)	30
Remarks on AVERAGE B (A. S. Radford)	34
Scottish Computer Education Report	41
Phased Training in Data Processing	45
Books and Articles - Short Reviews.	50

Editors' Notes:

This fifth issue of Computer Education incorporates several changes, some of which are obvious to the reader. The editors will welcome any comments from readers on either the contents or the layout of the bulletin. We plan to publish three issues regularly each year, one at the start of each term.

The typing, duplicating and assembly of the bulletin is now done by the staff of the Enfield College/English Numbering Machines Computer Centre, to whom we are grateful. At this issue the circulation has reached nearly 350, which represents a considerable amount of routine work. With a widening of our readership we hope to see an increase in the number of articles suitable for publication.

Computer Education Advisory Centre

Several readers of this journal have suggested that there is a need for an advisory service in the computer education field. This growing requirement was brought home to us one week in June, when one of the editors was approached by three colleges for detailed advice on the choice of computer equipment. We have decided that the editorial office of Computer Education will be available as a clearing house for colleges or schools who need advice on the various aspects of computer equipment, courses or staff. The editors are already in touch with college and school staff who have experience in this field and now wish to assemble a list of those who might be willing to devote some time for this purpose.

The staff at the Computer Advisory Service of the Ministry of Technology has been greatly strengthened during the last few months. Any educational establishment which plans to buy a full-scale digital computer should approach this group at an early stage. However there is much more to computer education than a digital computer and no substitute for hard experience in the educational field. This is where the Computer Education Advisory Centre fits into the picture. We are very willing to put a college or school which needs advice in touch with the appropriate expert in some other educational establishment. Some of this assistance may be given by letter, but we believe there is no substitute for a face-to face session, perhaps lasting up to one day, in this complex field. This implies that those requesting advice must be prepared to pay suitable travelling and accomodation expenses.

We are, therefore, awaiting letters from those of our readers who are willing to act as occasional advisors and requests from those who feel the need for a second or third opinion.

In closing we should mention that two technical journals have attempted to help in the choice of computing equipment. Every two or three years "Control" (28 Essex St. Strand, London, W.C.2.) has published a tabulated review of digital or analogue computers. Also "Educational Electronic Equipment" (31 Percy Street, London, W.1.) publishes reviews of small digital computers and analogue and hybrid computers, which are up-dated annually. Copies of the latest computer reviews from this second journal are available from the editor of Computer Education.

Computer Education Group - New Committee (May, 1967)

Chairman:	Dr. Jackson.	
Vice-Chairmen:	Mr. Lovis, Mr. Peacock	
Secretary:	Mr. Conway,	Staffordshire College of Technology, Beaconside, Stafford, Staff.
Treasurer:	Mrs. P. Jackson,	Staffordshire College of Technology.
Editors:	Mr. Abrahams, Mr. Conway.	Enfield College of Technology, Queensway, Enfield, Middx.
Ordinary Members:	Mr. Chamberlain, Miss D. Law, Mr. Friis, Mr. Smith, Mr. Stubbings, Mr. Thompson.	

New Constitution of Computer Education Group

1. The Group is established in accordance with aims initially laid down. The name of the group is "The Computer Education Group of the British Computer Society".

The group is an informal co-operative organisation for teachers (and others) interested in the use of computers in teaching in educational institutions. The group is particularly interested in the activities in schools, colleges of education, colleges of commerce, colleges of further education, colleges of technology and technical education.

The objects of the society are:-

(a) To spread knowledge in the field.

(b) To give aid to one another concerning problems:-

(a) in establishing courses of all kinds;
(b) in setting up equipment;
(c) in contacting industrial and other users;
(d) in contacting specialist lecturers.

(c) To make it possible to discuss one's problems, particularly with respect to the teaching associated with computing and computers.

(d) To provide a body which can approach other organisations with authority.

The society's activities will be:-

(i) One annual meeting held in the first week of May each year,
(ii) As many other meetings as the Committee think desirable,
(iii) The production of a bulletin which will be a collection of contributions from members of the group,
(iv) To further activate the spread of the use of computers in education at all levels.

Membership of the group does not involve membership of the B.C.S. There will be a form of institutional membership open to educational institutions which involves an annual subscription of £1. Ordinary annual membership subscription is 10s. 0d. Subscriptions due at the A.G.M. Subscriptions paid by new members joining between January 1st. & the A.G.M. will be accepted for the following year unless they specify otherwise.

The aims can be amended only at the A.G.M. by a majority vote.

2. There will be a central committee consisting of :-

Chairman,
2 Vice-Chairmen,
Secretary,
Treasurer,
Editors of the Bulletin,
Six ordinary members,
A representative of the Education Committee of the B.C.S.,
Ex-officio members of the Education Committee of the B.C.S., the Chairman, Vice-Chairmen and Secretary,
Branch representatives.

3. The Committee should at all time endeavour to be representative of all the various educational interests. Members shall offer themselves for re-election after three years.

4. The Chairman, 2 Vice-Chairmen, Secretary, Treasurer and Editor will be elected every year.

5. The Committee shall have the power to co-opt members.

6. Voting at the A. G. M. shall be restricted to members (1 vote) and institutional members (2 votes) present.

7. The formation of local branches is to be encouraged. Their function will be to organise meetings and activities in their area.

8. An honorary auditor will be elected by the A.G.M. and the accounts will be audited every year before April.

9. Notice of the A.G.M. shall be given in the Bulletin preceding the meeting. All items for inclusion on the agenda must be submitted in writing to the Secretary at least fourteen days before the A.G.M.

Aims of Computer Education Group

(1) To spread knowledge in the field of computing in education.

(2) To give aid to one another concerning problems:-

(a) In establishing courses of all kinds,

(b) In setting up equipment,

(c) In contacting industrial users,

(d) In contacting specialist lecturers.

(3) To make it possible to discuss one's problems, particularly with reapect to teaching associated with computers and computing.

(4) To provide a body which can approach other organisations with authority.

The group is an informal co-operative organisation for teachers (and others) who are interested in the uses of computers in teaching in educational institutions. The group is particulary interested in the activities in schools, colleges of education, colleges of commerce, colleges of further education, colleges of technology and in technical colleges. The strength of the group will lie in the activities of all its members and their willingness to communicate with one another.

The group's activities are :-

(1) One annual meeting held on Friday in the first week of May in every year.
(2) As many other meetings as the Committee think desirable.
It is also envisaged that there will be meetings in many areas of the country so members can get together and help each other.
(3) To produce a bulletin. This is the members medium of communicating with each other (Please use it !)
(4) To actively spread the use of computers in education at all levels.

Membership of the group involves a subscription of 10/- to cover the cost of producing the bulletin and circulating information. It also involves the willingness to help other members by pooling your experiences.

Subscriptions should be made out to 'The Computer Education Group (B.C.S.) and sent to D.E. Conway (Sec.), Staffordshire College of Technology, Beaconside, Stafford.

There is a school membership which is £1 and this entitles the school to 4 copies of the bulletin (and other literature) and to 2 votes at the A.G.M.

AIM OF THE BULLETIN

This publication is for the service of YOU, the READERS. The more material YOU provide the more successful it will be. Any ideas for series of articles will be welcome. Any contributions, big or small, should be sent to the Editor at the Computing Centre, Enfield College, Queensway, Enfield. It is intended that we should use the bulletin to help all our members. To this end we are going to set up a problem section which will function in two ways. Enquiries should be sent to the editor who will try and find some authority to provide the answers. Unless otherwise requested, the problem and answer will appear in the bulletin. If the enquiry is for general information then it will be published with a request for help from our readers.

We also intend to include in the bulletin references to books and articles of interest and, if submitted, reviews of the same.

There will be similarly, list of equipment in use and descriptions of success or otherwise. Difficulties and snags encountered in use of equipment would also be welcome so that readers do not have to suffer and can benefit by other experience.

We also want to provide information relevant to examinations and courses being provided in the field of computing.

To the same end we are going to give problems which we want to get members and their students to submit answers. The best answers will be published in the bulletin. All contributions for the bulletin should include the writer's name, address and affiliation. They can be hand written and in any form convenient for transit to the editor.

WE NEED YOUR CONTRIBUTION!

A few lines on your work or experience will be most welcome.

Letters to the Editor.

Enfield College of Technology
Queensway, Enfield,
Middx.
TEL;01:804:8349
Computing Centre.
May 10th.,1967

Dear Sir,

With reference to the letter written by J.T. Pye in the last issue of the B.C.S Computer Education Group Bulletin, may I comment on the choice of Computer languages to be taught to his sixth form students.

As Mr. Pye states, there are two basic high level mathematically orientated computing languages, Algol and Fortran. My first computing language was Autocode, but this only served as an introduction to Algol. After a years programming in Algol I decided to learn Fortran and have continued to use Fortran to this day.

The statements of any computing language must be concise, but they must also retain a palpable association to the logic of the problem they represent. The language must do the work for the programmer, otherwise the object of the computer is lost.

My main objection to Algol, is that it lacks the inherent simplicity in the program format, that is so patent in Fortran. Fortran statements are simpler than their Algol equivalents, because descriptive words have been removed, these words are superfluous to the understanding of the statement. It would be dangerous to remove all words and replace them with symbols such as "G" for "GOTO" and "W" for "WRITE", this would defeat the purpose of high level compilers.

Perhaps it would be useful to compare a few Algol statements and their Fortran equivalent

1) IF A B THEN GOTO L7' (ALGOL)

The Algol language assumes that if this condition (A B) is not detected then control is transferred to the next instruction. In Fortran the "Logical IF" statement is simpler.

IF (A. LT. B) GOTO 7 (FORTRAN)

The "Arithmetic IF" is even more flexible

IF (A-B) 6,7,8 (FORTRAN)
i.e. if (A-B) = -ve GOTO 6
" " = 0 " 7
" " = +ve " 8

2) "DO" statements are too verbose.

FOR K:= I STEP J UNTIL L DO (ALGOL)

This can be reduced to:

DO N K = I ,L ,J (where N is a statement no.) (FORTRAN)

Losing the words "FOR ,STEP, UNTIL" has not destroyed the meaning of the "DO LOOP".

3) PRINT £&L1?N,£S6??A; (ALGOL)

In Fortran

```
WRITE(3,1) N,A
1 FORMAT (1HO,16,6X,F10.2)                    ( FORTRAN )
```

Here we find the "WRITE" statement specifies the output device (3), the FORMAT statement no. (1) and the information to be written (N,A). The actual layout of the data is specified by the "FORMAT" statement which contains line feed, integer field width, spacing and floating point field width information.

There are other examples but the three cited I hope will illustrate my reasoning. I am sure it will be noticed that the FORTRAN equivalent for PRINT is longer than the ALGOL statement. This is true but the combination of "WRITE" and "FORMAT" statements enables the programmer to specify exactly what he requires, simply and logically.

If one analizes the arithmetic statements of the two languages there is very little to choose between them; the significant differences occur in the logical control, assembly and I/O statements, and here I feel FORTRAN is clearer.

I agree with Mr. Pye in that only one programming language should be taught in the sixth form college, and I think he will find that ALGOL will slowly disappear from the programming scene within a few years. At Enfield College of Technology we are teaching students FORTRAN with great success; it is a language quickly learnt and remembered, and extremly flexible in its application to various problems. Whilst I am on the subject of programming languages may I comment on the article devoted to finding an introductory programming language for students.

Apart from teaching FORTRAN at E.C.T. we are also teaching a simple language called "WORDCOM", this is a language created by Honeywell Controls Ltd. and can be used on their computers. It has been carefully designed to help the student grasp the ideas behind addressing, indexed addressing, instructions, control transfer and many other important aspects of programming. The entire language is created from approximately two dozen instructions, that have been reduced to obvious mnemonic codes;

i.e.

```
ADD  =  ADD
SUB  =  SUBTRACT
STA     STORE ACCUMULATION
JMP  =  JUMP

    etc.
```

The assembler language simulates a computer having a memory size of 350 words, which is more than adequate for student programs. The principles of the language are soon understood by students and the majority will write programs.

This language or any other instructional language is not wasted when the student progresses to FORTRAN, even though the foundation of the languages are so different. The instructional language has provided a useful insight into the functioning of a computer, and illustrated the " ART " in programming and above all, that it is the programmer who does the thinking.

Yours faithfully,

J.A.Vince.
Computing Centre.

ALGOL v FORTRAN

Whenever two computer experts get together it is almost certain that they will discuss, if not argue about, the superiority of Algol over Fortran or vice versa. I am in a very interesting position with regard to the Algol/Fortran controversy as I have not only changed sides from Fortran to Algol but also have been responsible for training, in Fortran, a large number of teachers who now believe Fortran to be a better language than Algol. Indeed, I wonder how many of them are reiterating my own earlier arguments. Had I been writing this note two years ago I should have said without any doubt that Fortran was a better language for teaching purposes and would have given the following reasons:-

1. When written it looks more like conventional Algebra.

2. It is less sophisticated than Algol and therefore makes fewer demands of the student.

3. It does not syntactically enable or cause the students to create such complex expressions.

4. It is more widely available because virtually all American computers have Fortran compilers and so do many British Machines.

Similarly, my arguments against Algol would have been:-

1. It requires a lot more writing and hence a lot more punching to write an equivalent Algol program to a given Fortran program.

2. It is logically a very sophisticated language and therefore likely to be very difficult for the student to absorb.

3. It is defined internationally but the international definitions completely lack in input/output instructions (vital to any computer language).

4. An Algol compiler is a very complex and sophisticated program and therefore one is unlikely to get a compiler for on a small computer.

5. It is not so widely available as Fortran.

I was forced into the situation (I am glad to say) of having to teach Algol to my students because the computer which we installed here had the Algol compiler made available, by the Manufacturers, before the Fortran compiler. I have now, for almost two years, taught Algol to people of widely varying abilities. With this experience behind me I now regard Algol as a better teaching language than Fortran. In view of my experience I have now demolished my own arguments against Algol as follows:-

1. The verbosity of Algol not only makes it lenghty to write but also makes it far more intelligible to the student. Because the language so closely resembles the English language, the student is encouraged

to explore more varied facits and constructions than he would otherwise. Admittedly, he occasionally makes mistakes but nevertheless this exploration by the student is educationally desirable.

2. The students are capable of writing more sophisticated programs than I had expected. Therefore they need only study in a limited subset of the Algol language and therefore what they do study need not conceptually be more difficult than Fortran. I add this last note to act as a carrot for teachers who themselves are a little worried about Algol.

3. The input and output instructions can easily be translated from one hardware representation to another and bodies, such as the European Computer Manufacturers' Association, have defined Algol input/output facilities. I might add that the Elliott Algol input/output system is very good and far superior to any Fortran input/output system which we have used. Because of its international acceptance Algol is the standard language for published programs and, therefore, there are available, to the teacher and student, many examples.

4. It is interesting to note that for the Elliott 903 Computer, the Algol compiler is smaller and slightly faster than that for Fortran. This is because in view of the fact Algol is so logically designed that it is capable of having applied to its compilers, more sophisticated compiler writing techniques.

5. This argument is still true to some extent but now more and more manufacturers are offering Algol as a standard compiler on their machines. Most Universities and Technical Colleges where schools are initially getting their computer experience, will have Algol compilers. I am now certainly a convert to Algol. Algol is unlikely to last forever, but there is an ever increasing number of good books on the subject and because of the massive investment in software and user programs, Algol compilers are going to be available for many years yet.

Having written all the above, may I state one fact which I believe to be more important than all that has gone before. This is that it is better to learn to program a computer in any language than not to learn to program at all and therefore I would strongly urge teachers who have any facilities available to them to make use of them in whatever language they have to use. Bearing in mind that once a student has learnt to write programs for a computer in any language he will find it extremely easy to pick up a new language.

W. Broderick,
Royal Liberty School,
Romford,
Essex.

Physics Dept.
Omskirk Grammar School,
Ruff Lane,
Ormskirk,
Lancs.
5:1:67

Mr. Conway,
Editor,
Bulletin of Computer Education

Dear Sir,

Recently I have been working on a design for an electrical binary subtraction unit. Being uncertain of the correct way to subtract by taking the complement, after a little experimentation I found a method which gave the answer, without the need for any correction, as in other methods.

The method consists of,inverting the first number, then adding in the second and inverting the answer. An example is shown below.

To take 1000 from 100,

Take	1000	
Invert	0111	
Add in	100 +	Correct Ans = 100
	1011	
Invert	0100	

On discussing this with my physics teacher we found that this method is theoretically correct. Inverting the binary number is the same as taking the complement. An algebraic example in the decimal system is shown below.

x - y

take complement of x	= 10-x
add in y	= 10-x+y
take complement	= 10-(10-x+y)
Result	= +x-y

We also found that the number need not be as large as the binary bit, (necessary in the usual method) in order to give the correct answer. This removes the need for any device to control the section of the bit to be used, or a device to make corrections for this when necessary.

e.g. 0,000,111 - 0,000,011

Invert 1st number	= 1111000
add in	0000011 +
Invert	1111011
Giving Ans	= 0000100

With a little core in design this system can give a fair amount of electronics, by removing the need for correcting circuits. I would be grateful to know if this is in

fact a new method, and wonder if any of your readers know of it being in use at the moment. If it is a new idea I think that it could well prove to be of use to some of your readers.

Yours sincerely

M.E. Haughton

Telephones
Masters - 368011
Bursar - 2016

St. John's School,
Leatherhead,
Surrey.

28th. April, 1967

PD/5/S.

D.E. Conway, Esq.,
Staffordshire College of Technology,
Beaconside,
Stafford.

Dear Mr. Conway,

The arrival the other day of the latest issue of "Computer Education" reminds me once again that I haven't yet written anything for the journal. Unfortunately our term starts here to-day, so this isn't the time to go into much detail, but I thought I would at least tell you briefly what we have been doing here since last April!

On the hardware side first, I have continued to extend the scope of 'SUSAN', the simple digital teaching aid machine that I started some two years ago now. Since the article I wrote last year (a copy of which I believe I sent you - I hope!), I have modified the circuits in a big way. Not only will she perform all the functions as a binary adder as described in the article, but now she can be switched over into a serial machine. Using only an 8 bit word and three shift registers, she can add subtract or multiply. All three can function at a clock speed of about 1 Hz, and the addition and subtraction functions will also function at a 'so-called fast' speed of about 250 Hz. The big question that I have now got to answer is "what next?"! I may just stop here, and perhaps write up what I've done so far; I may continue the hardware and build the circuits for division - I've got all the logic worked out! Or (and I think this is probably the most likely) I may try to convert the present machine into the central arithmetic unit of a larger system by designing and building the necessary storage, control, input and output facilities. This obviously is going to be quite expensive in time and money! I do have an old Flexowriter already which might be converted to act as a suitable input and output medium. So much, though, for 'SUSAN'.

Still on the hardware but probably not of so much interest to you (if you remember, I'm a physicist and not a mathematician!), I have done some work on analogue computing, and with the help of a local firm, have designed a series of simple physics experiments for 'A' level students to illustrate the techniques of such computing. Some details of this are going to appear in Physics Education, but I'm not sure when.

Lastly - software. Since last September I have been running a course on digital computer programming for our Sixth Formers as an 'option subject'. All the VIth form boys can choose one or two such subjects (usually two periods a week), and I started off with 25 boys including a few non-scientists. We worked with Pegasus Autocode, mainly because we were extremely fortunate to be given time for free on one of the Pegasi machines at the British Aircraft Corporation at Weybridge. The numbers dropped, (boys leaving, pressure of 'A' levels etc.) but we continued this work until the end of February with some 15 boys. In March, we had a short condensed course with Ronsons on commercial data processing ending up by interrogating one of their files on the IBM 1440 system they have. We all found this extremely interesting and useful, mainly publicity given now-a-days to this use of computer time.

Now this term we are going to do some work with Algol and the University of Surrey computers, probably their Elliott 503. I've been doing some work on this during the holidays and at last beginning to realise at first hand what is meant by a 'high level language'! We'll probably be down to 9 or 10 boys this term.

This connection with Surrey University is one that I personally am extremely anxious to foster, for it will be extremely useful for us when the University moves down to Guildford, starting next year. Their computer system hasn't been decided as far as I know, but there is a slight possibility that we may be able to get a telex data link with it once it is fully operational. It is an interesting possibility though. Do you know of any other schools (other than Hatfield School) making similar plans? I remember the bit in one of the Bulletins about the Cornish schools, but that didn't give any indication of what was really happening or when.

I said this was going to be brief, and as usual I seem to have gone on for some time! However, my last point. We are coming up against the usual problem of what problems to set the kids which they stand a reasonable chance of tackling successfully and which are still meaningful to them. I know this has come up in several of your Bulletins, but any ideas will be more than welcome. In return, can I offer one idea which we borrowed from B.A.C. If one imagines a metal plate that has already cost a great deal of time and money to machine and shape, and to this circular piece of equipment muct be bolted with 3, or 4, or 5 etc. bolts. The coordinates of the centre of this circle will be given (either rectangular or polar) with reference to one corner of the metal plate. Calculate by two completely independent methods the coordinates of the holes which must be drilled to take the bolts (oh - one is also given the radius of the circle of course!). There must not be any mistake, and the holes must be placed symmetrically around the circle. We haven't completed the task yet, but others may care to try it.

Yours sincerely,

P.E.Dutton.

CONFERENCES AND COURSES

The British Computer Society.

DATAFAIR 67.

Why Computer Education in Schools?

26 September 1967.

CHAIRMAN: Mr. M. Bridger - Chairman of the Education Committee of the British Computer Society and Head of the Department of Mathematics, Computing and Statistics, Leicester Regional College of Technology.

10.00 a.m. Introduction - Dr.J.J.B. Dempster, Chief Education Officer for Southampton.

TODAY'S PIONEERS.

10.15 a.m. Why Computer Education in Schools? - Dr. E.D. Tagg Chairman of British Computer Society Working Party on Computer Education in Schools and Computer Careers and Head of Department of Mathematics, Oundle School, Northampton.

10.30 a.m. Coffee.

10.45 a.m. Analogue Computers in Schools - Mr. M. Meredith, Towyn School.

11.00 a.m. Computer Building in Schools - Mr. A. Wilkinson, The Grammar Technical School, Spennymoor, Co. Durham.

11.20 a.m. A School Computer Department - Mr.W.R.Broderick Head of Computing Department Royal Liberty School, Romford Essex.

11.40 a.m. Computing for Mathematics Pupils - Mr. G. Goldstein University College School, Hampstead.

1200 noon Computing for Arts Pupils - Mr. D. Pegg, Bedford School, Bedford.

12.20 p.m. The Role of the College of Education - Mr. M. J. Farringdon Lecturer in Mathematics, St. Luke's College, Exeter.

12.40 p.m. Lunch

WHAT DOES THE FUTURE HOLD?

1.45 p.m.a) Multiple On-Line Equipment for Educational Use - Dr. S. Summersbee, Department of Computing, Cybernetics and Management, Brighton College of Technology.

2.15 p.m.b) A demonstration of a real-time system for performing numerical manipulations from a number of remote teleprinter consoles simultaneously.

2.45 p.m.c) Computer Aided Instruction (C.A.I.) - a report on current trends introduced by Mr. M.J. Cooper, Manager Education Marketing, I.B.M. United Kingdom Ltd.

3.15 p.m. Tea.

THE BRITISH COMPUTER SOCIETY AND THE

COMPUTER EDUCATION GROUP

3.30 p.m. The Computer Education Group - Mr. D. Conway Secretary of the Computer Education Group and Principle Lecturer in Computer Science at Staffordshire College of Technology.

3.45 p.m. Computer Careers and Computer Courses - Miss D. Law, Manager, Education Liason Services, International Computers and Tabulators Limited.

4.00 p.m. General Discussion.

4.45 p.m. Closing Remarks (Mr. M. Bridger)

5.00 p.m. Seminar Ends

NOTE:

The Elliot Mobile Computer classroom with a 903 computer will be parked outside the lecture theatre and can be visited before and after the seminar and during the coffee, lunch and tea breaks.

IFIP EXHIBITION 68

International Exhibition of Electronic

Data Processing

Organised by the International Federation for Information Processing to run concurrently with IFIP Congress 68, will be an exhibition of the latest equipment and ideas from one of the world's fastest growing industries - electronic data processing.

Although still over a year away, the response to this exhibition has been such that already 90 per cent of the space has been provisionally booked. Open from the 5th to 10th August 1968 at the Waverley Market Exhibition Hall, Edinburgh (Scotland), the exhibition will feature the products of all the major computer manufacturers, provide demonstrations of the most advanced electronic data processing concepts, and show current applications of prominent world business and scientific users.

The Waverley Market Exhibition Hall, centrally located and within easy reach of all the major Congress locations, will provide approximately 30,000 square feet (2,800 square metres) of exhibition space.

Special facilities will also be available for the display of technical books concerned with the information sciences.

R.D. D'Urban,
Organiser, IFIP Exhibition 68,
109 Kingsway,
London, W.C.2.
U.K.

Tel: HOLborn 6233.

} will supply further information

International Federation for Information Processing (IFIP)

IFIP is an international organisation and was formed in 1959 to provide a link between the various information-processing communities of the world to assist in the development of the information sciences. Since the UNESCO sponsored international conference in June 1959 at which IFIP was created, IFIP has held triennial congresses and exhibitions, and at least two specialist conferences each year, for the study of all aspects of information processing and computing.

****

WIGAN AND DISTRICT MINING AND TECHNICAL COLLEGE

MATHEMATICS DEPARTMENT

N.W. Regional Branch of the Computer Education Group

Branch subscription:

Without "Computer Education" Bulletin 5s.
With "Computer Education" Bulletin 15s.

Proposed Programme:

1967-8

October:	At Wigan:	Speaker from National Computer Centre, Manchester.
November:	At Blackburn:	Writing and running a simple program.
December:	At Wigan:	Speaker, - Steps in the computer solution of an Industrial Problem.
February:	At Wigan:	Speaker, - Historical development of Computers.
March:	At Widnes:	Telex links and programming for schools.
April:	At Wigan:	A.G.M. Use of Calculating Machines in Teaching.

R. H. PEACOCK
Branch Secretary

Inaugural meeting of East Midlands Branch.

4th October 1967 7 pm. at City of Leicester College of Education, Scraptoft, Leicester
Will all local members please make a special effort to attend this meeting.

ENFIELD COLLEGE OF TECHNOLOGY

COMPUTING CENTRE
TEL: 01-804-8349

The College proposes to run five courses during 1968, specifically for teachers who wish to introduce a computer education subject into the curriculum.

The College has been active in this field for some time, and experience from the four courses run during the current session has suggested that a duration of four days and timing at the beginning or end of the spring and summer terms causes least interference with school routing.

The enrolment for each group will be limited to 20, and the College will issue a list of suitable hotels to any teacher requiring residential accommodation convenient to the College.

FORTRAN PROGRAMMING.

E 1 Tuesday 2nd January - Friday 5th January.
E 2 Monday 29th April - Thursday 2nd May.
E 3 Monday 1st July - Thursday 4th July.

Lecturers: Mr. N.E.Bowker, Mr J.C.Williams (Enfield College)

A thorough treatment of Fortran D will be given, with ample opportunities to write and run programs on the College computer.

LOGIC TUTORS AND THEIR APPLICATION.

E 4 Monday 25th March - Thursday 28th March.

Lecturer: Mr. N.E.Bowker (Enfield College)

These lectures and practical sessions will deal with the algebra of logic, and show the value of logic tutors in modern subjects at school level.

COMPUTER TEACHING AIDS.

E 5 Monday 1st April - Thursday 4th April.

Lecturer: Mr. N.E.Bowker (Enfield College)

This course will introduce the range of teaching aids developed in the College to assist the teaching of logic and analogue and digital computing. Ample opportunities will be given for practical work with the equipment.

The B.C.S. Working Party No. 4 are investigating the following subjects and would like to contact anyone interested. If you have an interest the editor will be pleased to pass your name and address on.

BRITISH COMPUTER SOCIETY EDUCATION COMMITTEE WORKING PARTY NO. 4

TOPICS FOR INVESTIGATION

1. A survey of the educational work in progress in secondary schools related to data processing.
2. Curricula for school work.
3. Equipment and methods of practical work.
4. Textbooks and exercises.
5. Education of teachers in data processing.
6. Impact of computers on other school subjects e.g. mathematics, economics, sociology, etc.
7. Computer Education for all. Is this necessary or desirable? If so, to what extent?
8. How can teachers of other subjects than mathematics be interested in computer education.
9. The computer as an educational facility.
10. A perspective view of computers in school education. Freedom and clarity of thought as more basic than a national effort to lead in the computer field. Why abler students do not prefer thin sandwich courses.
11. Christmas holiday lectures.
12. Progress reports on present and future developments in possible computer facilities for schools.
13. Public relations with (a) teachers, present and future, (b) pupils, (c) computer makers and users, (d) other educational bodies.
14. Careers in computing - keeping aware of the changing needs and opportunities.

Report No. 1 of the Working Party

for

Education of Scientific Programmers

of the Education and Training Study Group

of the Manchester Branch of the BCS/

This paper is a collection of thoughts and opinions accumulated over six years' lecturing to scientific programmers on DEUCE, KDN2, KDF7, KDF9, Algol, Fortran and Introduction to Programming. The reader is asked not to confuse statements of fact with statements of opinion, which latter, if they provide points for discussion will have served their purpose.

Six years ago there were still many people being trained to program first generation computers. This training involved Machine Code rather than some simplified User Code, with all its attendant intricacies and gimmicks. The programmer very often tested his own programs and so a training course had to provide practical experience on the operation of a large complicated console.

This type of programming attracted a particular type of mind and the aptitude for programming was not directly related to intelligence. It was the type of mind which found great satisfaction in the sort of puzzles found in the quality Sunday papers, only there was a difference. One can always give up in attempting a puzzle, and wait for the solution to appear in the next addition, whereas in programming a solution must be found.

The rigid discipline so imposed created an intimate relationship between the programmer and his program, with the result that the programmer regarded his completed work with the same pride as a master craftsman might regard his. One of the most difficult tasks in programming was to ammend a program written by someone else, even though it may be very well documented. It was very often easier to re-write the program.

With the coming of second generation machines, the increased complexity of the computers was partly relieved by mnemonic User Codes. The effect of User Codes was to simplify coding, and this was accompanied by a standardization of techniques which removed from programming some of its pioneering flavour. The older hands may perhaps regret the passing of the early days, but there is no doubt that any simplification of the programmer's task immediately gives a wider field from which programmers may be selected, and that is essential when the industry is expanding at its present rate. Many establishments used to select their programming staffs from those with first class degrees upwards, whereas now it is not uncommon for school leavers with GCE A levels to come on programming courses.

From the training point of view, these younger students are very often the most responsive. This is probably due to the fact that they are still attuned to intensive learning, and are further stimulated by the new concepts. Young graduates, while displaying ability and greater bredth of background than school leavers, very often lack that spark of enthusiasm. This is understandable after emerging victorious from the rigours of a three year degree course.

A student over tha age of 30 is almost invariably slower than his younger companions. He finds the new concepts difficult to grasp and to apply, but he works more conscientiously and steadily until he has a complete mastery of the subject. This can be a source of great satisfaction to the Instructor.

The usual pattern of a programming department these days is for there to be a hierarchical structure from a Chief Programmer at the top to a Programming Assistant, or Coder at the bottom. A person's place in the hierarchy, and subsequent promotion within it will depend on a combination of factors based on his previous background, his programming ability and his experience.

These people are professional programmers who have undergone intensive training lasting several weeks. This training has to be paid for, as has their travelling and hotel accommodation, and for the duration of the course the students are paid their full salery. All this represents a considerable investment on the part of the employer. There are computer users however, whose use of a machine would be insufficient to warrant such expenditure on training. It was with these people in mind that the computer manufacturers developed the early simplified programming schemes, Alphacode, Mercury Autocode etc. These schemes are very uneconomical in terms of machine usage, being very much slower in speed than Machine Code, but in terms of training and program writing they represent a considerable saving. Thus the occasional user who wants a single machine run will find it cheaper to attend a short course to learn the scheme quickly, and write his program in half a day, despite the machine time used.

On the other hand the man with the big program, which will be run frequently, will be wise to obtain the services of a professional programmer who will write a program with optimum coding to save as much machine time as possible. This will effect savings in the long run despite the time taken to write and test the program.

The autocodes are being replaced by Programming Languages, particularly Algol and Fortran. The latest implementations of these are remarkably efficient in their use of the machine, which disposes of the main criticism of the earlier simplified coding schemes. Furthermore, Algol and Fortran, having a linguistic form, are easier to remember because of thh resultant fluency of expression.

Quite apart from the economic considerations, a scheme such as Alphacode is remarkably easy to teach, sufficient to write a large program can be taught in half a day. As a pure teaching medium Alphacode is ideal as it contains all the important programming concepts such as loops and counting, and Instruction Modification. Through ignorance one can only assume that the same advantages are to be found in Mercury Autocode. But to teach Alphacode in vacuo can be a waste of time, if the students are complete newcomers to programming.

Experience in lecturing on several different machines has revealed the fact that there are certain topics common to all courses, and to labour these topics on different courses can be a misuse of valuable time. The common features on these courses have been used to formulate a course called Introduction to Programming, and students with no previous experience are expected to attend the Introductory course before going on the appropriate machine course. It has been found that the course provides an excellent springboard to matter what the subsequent course. Valuable time is saved on these subsequent courses because elementary concepts have already been grasped, jargon is immediatly comprehended, and the student has a greater sense of confidence. The Introductory course is also useful for giving an insight into computing to people who only want a familiarisation, and have no intention of undergoing further trainning. Thus as it stands the course serves two purposes at once.

In essence, the syllabus includes a simplified explanation of the working of a typical computer, a description and demonstration of peripheral equipment, a brief description of hardware, binary and octal arithmetic and flowcharting. (See appendix) Use is made of the examples of flowcharting to illustrate typical programming techniques such as iteration etc. Taking this syllabus as a hard core, it is possible to mould the course to specific requirements by careful shift of emphasis, or perhaps by augmentation. Thus by selecting the examples and exercises, different branches of technology can be catered for and the course may be made suitable for sixth formers or teachers by including programming practice in an autocode, with actual machine runs. This has the effect of making the course really come alive to youngsters; it doesn't matter how trivial the exercise may be as long as the student has some tangible results from the output of the computer.

Although one's experience is mostly with the concentrated full time course, one might venture to suggest that this is a more economic form of instruction than one or two lecturers per week spread over several weeks. No time is lost in recapitulation at the beginning of each lecture, and greater flexibility with the time table can be exercised to suit the collective ability of a particular group of students.

No matter who attends the Introductory course, or whatever version of it is given, the aim is the same, to show that there is nothing magic or frightening about computers, and to get the students to rethink their problems in a more analytical way so that they may form a better judgement of the feasibility of using computers in their own disciplines. Reliance is placed on the student's ability to apply his newly learned information to his own sphere of work, and this is necessary as there is a limit to the number of applications with which any lecturer can become acquainted.

This course is of great value anywhere where computers need to be introduced, whether it be to experienced professional engineers or to school children. From here it is a very short step to an Algol or Fortran course where the concepts can be seen being put into practise, and the structure of the language can be gone into without preamble. Here is the means to get people computer minded, and the benefits will be felt more and more recruitment of computing personnel continues to increase.

COMPUTING CENTRE AT ENFIELD COLLEGE

by J.R. Abrahams.

Introduction.

The Computing Centre at Enfield College of Technology was opened officially on April 26th., 1967, by Mr. John Mackie, M.P. and the Rt. Hon. Iain Macleod, M.P. This centre has a number of features which should be of interest to many in the educational field. The computing activities at Enfield College developed out of the original electrical engineering department due to the large proportion of electronics manufacturers in the area. This unusual background, compared with the typical growth of computing from the mathematics department, gives a certain flavour to the installation. On the other hand the present academic work of the centre now includes a large proportion of business mathematics and commercial data processing.

Computer Equipment

The computing facilites at Enfield College are conveniently accomodated on two floors of one wing of the main building, in a total floor area of about 4,000 square feet. This includes office space for both the teaching and operating staff.

The digital computer is a Honeywell 120, which has been operating since October 1966. The H 120 is almost as fully equipped as possible, with a core store of 24,000 characters and the following peripherals:

Four magnetic tape units (13.3kc/s)
Card read/punch unit (400 cards/min)
Line printer (450 lines per minute)
Paper tape reader (600 characters/sec)
Paper tape punch (150 characters/sec)
Digital Plotter (Calcomp 565 on-line)
Console Typewriter

The Honeywell Computer is operated on a "closed-shop" basis, with a team of full-time operators. These operators are responsible for program reception and dispatch, and also for most of the data preparation word. At present the digital computer is operated for a 12 hour day and the staff of two shift leaders and seven operators would allow for extension to a full two-shift 16 hour day.

For our purposes at Enfield College we use the word "computer" in a fairly wide sense, to cover any machine from a simple desc calculator, through analogue and hybrid machines, to the full electronic computer. All this equipment is therefore located in the Computing Centre. We have a PACE TR-48 Analogue Computer, which which is now almost fully equipped, with three detachable patch-boards to give economical use of the actual computer. This 10v analogue computer is available on an open-shop basis, for any staff or students, and has been used on an interesting variety of problems including operational research.

The first stage of a hybrid computer to be delivered in four annual instalments should be in operation within a few weeks. This is the Lan-Hitachi 505 Computer,

which will grow to have 120 amplifiers and a digital section with a core store of 8000 words. Since the analogue section of the hybrid operates at 100V this computer will be used under staff supervision. Our desk calculator complement consists of four electronic machines, eight electro-mechanical calculators and sixteen simple mechanical machines. We intend to expand to twelve Friden electronic calculators as soon as possible. All these machines are freely available to students for a 12 hour day.

Teaching Aids

Over the past four years several members of our staff have been active in the design of teaching aids for computer education. Three of these aids are now in full-scale production and are used in many educational establishments, and so it is natural that we should have a good stock at Enfield.

The electronic logic tutor is extremely useful to show the principles of the digital computer, and other digital systems, both from the mathematical and electronic points of view. With the trend to large, remote computers the logic tutor is the nearest that most of our students will get to operating any digital hardware. We are very much concerned with a number of courses for computer engineering, at various levels, and the logic tutors are used intensively on these courses. Our Computing Laboratory now has twelve of the original Lan-Dec 15 logic tutors and ten of the newer more versatile Lan-Dec 20 units (which incorporate integrated circuits). For certain periods of the year the LD-15 units are available for loan to local schools.

We are great believers in the small analogue computer, both as a teaching aid in its own right, used by the lecturer and as a means of introducing the students to many computer concepts, particularly those of scaling and simulation. We now have ten Lan-Alog 4 computers and use each with a Telequipment S51T oscilloscope, so that we can teach a class of up to 20 students at once. This has proved to be a far better approach to teaching this subject than buying just one medium-sized analogue computer, for about £3,000.

Within the past year we have given four obsolete valve-equipped analogue computers (two Solartron and two Heathkit) to local schools. Although these machines are not very reliable electronically this has enabled these schools to do some useful introductory work.

Our computer teaching staff are all ready using two other teaching aids, which were shown this year at the Physics Exhibition, but have not yet appeared commercially. The first of these is the Lan-Dec 310, computer arithmetic tutor, which has three 10-bit registers and can perform all the basic digital computer operations. This unit has fairly large coloured indicator lamps for each part of the system, and can operate at slow or high speed. The Lan-Dec 310 can show binary or octal arithmetic, with single or double word lengths.

Another more complex unit is a table-top hybrid computer, with ten multi-purpose amplifiers, three multipliers, two comparators and thirty digital circuits. This small student-proof computer became necessary to introduce students to the techniques of hybrid computing, which is quickly coming into use in the engineering, bio-medical and operational research fields. It will also be useful for testing parts of programs for the large hybrid computer, to free time on the much more expensive machine.

Academic Staff.

From this coming September we shall have ten academic staff at Enfield who specialize in computing and data processing. As a temporary measure the Principal Lecturer is both computer manager and head of the academic section. There are four Senior Lecturers, and five lecturers. One senior lecturer is responsible for computer technology, particularly on the digital side, and is assisted by a lecturer in analogue computing. One senior lecturer and two lecturers will be responsible for the new intensive courses for Systems Analysis, which are sponsored by the National Computing Centre. A third senior lecturer, together with two lecturers, will be responsible for all instruction in computer programming and for staffing a full-time programming advisory service in the computing centre. The fourth Senior Lecturer has an appointment in Computer Education, which is thought to be the first of its type in this country. This post is for close liaison with all outside educational users of our computing facilities, including the provision of in-service courses for lecturers and teachers.

Educational Use.

The first aim of the Enfield Computing Centre is to provide a service for all staff and students of the college. Most undergraduate and research programs are now being written in Fortran, and several large projects are now being used in Cobol. It is also our intention to provide a computing service to other colleges and schools in the four boroughs of outer North London, an area with a population of over one million. Three Schools and three colleges are all ready sending us programs fairly regularly through the post. This is proving satisfactory as an interim measure for a light user, since we can offer a one-day turn-round.

The next stage will be the opening of off-line data links, equipped with Westrex teleprinters, to Southgate Technical College and Hendon College of Technology in October of this year. These links will be over full-time rented G.P.O. lines, and, due to the relatively short distances involved in North London, the rental does not appear excessive. We expect to open two more such links, one to a large school in Haringey and the other to East Herts College of Further Education, in January 1968.

Within Enfield College we hope to teach basic Fortran to all full-time undergraduates, regardless of their field of study. In one sense this will be "liberal studies for arts students." It seems likely that by mid-1968 about 1,000 students and school pupils will be learning computer programming each year through the Enfield centre.

Industrial Co-operation.

In common with most colleges of our type we are keen to develop industrial and commercial contacts at all levels. This co-operation has been carried a long way in the installation of our Computing Centre. The Honeywell digital equipment has been financed equally from the college and English Numbering Machines Ltd. In return for this capital contribution E.N.M. have a claim to two-fifths of the computer time in the twelve-hour day. In addition E.N.M. pay a monthly sum equal to

half the operators' salaries. From September the college and E.N.M. will also be operating a joint punched card installation, with eight I.B.M. punches, four verifiers and a high-speed card sorter. It is hoped to achieve a similar industrial participation in the hybrid computer installation. We are now finding that this close contact with a lively medium-sized firm is helpful for our staff, particularly in the business mathematics and studies courses.

Future Development

We expect the major development of the next two or three years to be the acceptance of reliable multi-access systems, with one hundred or more remote terminals. Several groups in this country are now working on the software for such systems, and we can foresee an increasing educational demand in our area which will soon justify such a system, costing perhaps £500,000. We hope to publish our detailed proposals within the next few weeks, and to include some details in the next issue of this journal.

"JACK-OF-ALL-TRADES"

J. Bradshaw B.Sc. (Econ). I.C.T. Ltd.,West Gorton.

Many questions have to be asked and answered before computing is introduced into the syllabus. Depending on circumstances more or less heart searching takes place. It is probably easier for an examining authority to introduce an advanced level certificate in computer science of a national diploma in mathematics and computing than it is the individual primary school teachers to reconcile the introduction of binary numbers with her code of professional ethics. Purpose and aims have first to be established.

At the higher, more specialised level of education purpose and aims are more clearly defined. The student has a choice of courses and subjects; he has previously shown interest and aptitude; his education can be "liberalised" by the introduction of a single subject. Certain assumptions, valid or invalid, are made about him. It is assumed for the purpose of teaching that occupationally he will become directly concerned with computers. He will be an engineer, or programmer, or systems analyst or designer; if he does not he will be the exception.

The teaching of computing to the senior arts student calls for a little more imagination. He may enter one of the traditional or newer liberal professions and come to have a passing aquaintance with comptuers, or he may become a commercial programmer and data processing manager. Again computers may play no part in his life of which he is ever aware. Nevertheless the senior arts student is mature enough to take an intelligent interest in the implications of "datamation".

It is in the sphere of secondary education that the teacher is faced with more possibilities, more decisions, more experiments, and more failures. What assumption can he make about his pupils? What should his aim be and to what purpose does he struggle to master the principles and practice of computing? It would be a great help if the teacher could assume a computer "jack-of-all-trades: someone who will one day need a basic knowledge of electronic terms; who will be able to understand technical literature; to learn to operate a console; to write (and read) a simple program; to use a flexowriter; to amend manuals; to complete forms; and to learn without being taught. The relevant questions are does this "jobbing builder" exist in the world of data processing; will he find a permanent place; and if so will his numbers increase?

There are computer manufacturers and computer users. The jack-of-all-trades is already known to the manufacturer; he may well become a feature of the user's establishment.

Such a jack-of-all-trades is the operator/diagnostic programmer who works with the commissioning engineer. He provides test program paper tapes and cards, writes, punches, and debugs his own short programs, operates the computer in test, writes reports, maintains a library, amends manuals and carries out numerous other tasks. He learns daily and can ignore no aspect of hardware or software. How much he learns and how quickly depends on obvious factors not least of which is, "what he knew when he started". Yet it is unlikely that he had formal pre-training. No advanced course caters for all aspects of his work, and had he specialised he would probably have sought a more specialist job. His prior knowledge of computing, if any, he gained at secondary school. He stands between the specialist and the layman. He may well by the mean at which the secondary school teacher can aim, if he emerges in sufficient numbers.

Many questions have to be asked and answered before computing is introduced into the syllabus. Depending on circumstances more or less heart searching takes place. It is probably easier for an examining authority to introduce an advanced level certificate in computer science or a national diploma in mathematics and computing than it is for the individual primary school teacher to reconcile the introduction of binary numbers with her code of professional ethics. Purpose and aims have first to be established.

At the higher, more specialised level of education purpose and aims are more clearly defined. The student has a choice of courses and subjects; he has previously shown interest and aptitude; his education can be "liberalised" by the introduction of a single subject. Certain assumptions, valid or invalid, are made about him. It is assumed for the purpose of teaching that occupationally he will become directly concerned with computers. He will be an engineer, or programmer, or systems analyst or designer; if he does not he will be the exception.

The teaching of computing to the senior arts student calls for a little more imagination. He may enter one of the traditional or newer liberal professions and come to have a passing aquaintance with computers, or he may become a commercial programmer and data processing manager. Again computers may play no part in his life of which he is ever aware. Nevertheless the senior arts student is mature enough to take an intelligent interest in the implications of "datamation".

It is in the sphere of secondary education that the teacher is faced with more possibilities, more decisions, more experiments, and more failures. What assumption can he make about his pupils? What should his aim be and to what purpose does he struggle to master the principles and practice of computing? It would be a great help if the teacher could assume a computer "jack-of-all-trades": someone who will one day need a basic knowledge of electronic terms; who will be able to understand technical literature; to learn to operate a console; to write (and read!) a simple program; to use a flexowriter; to amend manuals; to complete forms; and to learn without being taught. The relevant questions are does this "jobbing builder" exist in the world of data processing; will he find a permanent place; and if so will his numbers increase?

There are computer manufacturers and computer users. The jack-of-all-trades is already known to the manufacturer; he may well become a feature of the user's establishment.

Such a jack-of-all-trades is the operator/diagnostic programmer who works with the commissioning engineer. He provides test program paper tapes and cards, writes, punches, and debugs his own short programs, operates the computer or test, writes reports, maintains a library, amends manuals and carries out numerous other tasks. He learns daily and can ignore no aspect of hardware or software. How much he learns and how quickly depends on obvious factors not least of which is, "what he knew when he started". Yet it is unlikely that he had formal pre-training. No advanced course caters for all aspects of his work, and had he specialised he would probably have sought a more specialist job. His prior knowledge of computing, if any, he gained at secondary school. He stands between the specialist and the layman. He may well be the mean at which the secondary school teacher can aim, if he emerges in sufficient numbers.

My feeling is that his kind will increase in both small and large establishments. In the small D.P. unit the utility man and the "open" system suggest themselves. In the very large numerically controlled factories and installations of the future with their hybrid system complexes personal versatility will be needed to overcome structural rigidity.

The questions for the teacher to answer are what do I teach and how do I teach it? A knowledge of magnetism and electricity is a basis upon which electronic terms and meanings can be built. A comprehension of technical and semi-technical literature can be added to general and literary English comprehension. Some practice at soldering and wiring could supplement metal work and woodwork. Mathematics can be developed naturally in the directions of number systems, analysis and truth tables. Mechanical counting devices have a long history and a geographical basis. There is the problem of language: a Swedish or Arabic barrel on the line printer of a British or U.S. computer. It begins to look as though no subject nor teacher can escape. This suggests teaching method. Computing need not call for more specialist staff and more specialist teaching.

All that is needed is a little co-ordination and co-operation which ensures that no teacher overlooks the impact of the computer on his or her discipline. After all is not the secondary school teacher a jack-of-all-trades.

J. Bradshaw,
West Gorton.

"Computer Education"

Extracts from a talk given by Miss D. Law at a seminar entitled "People and Computers". This was arranged jointly by I.C.T.Ltd., and the Staff Management Association.

The mystique about computers and the "Doctor Who" concept of computers is dying out but not quickly enough: an aura of mystery mumbo jumbo and magic too often surrounds computers. Not enough of us think of the computer as an electronic filing cabinet cum slide rule which is as good a description of a computer as any other. As my colleague George Phelps indicated this morning if you get the input and the output under control one is home - home and dry.

Take the City and Guilds 319 and 320 courses. Nobody seems to be quite certain for what type of person these courses were originally designed. Was the 319 designed for a computer operator or for a junior programmer? For what sort of person is the 320 course designed? At no time does there seem to have been a job specification in anybody's mind. These courses, too, are far too mathematical for the normal person working in a commercial computer set-up. For people working in scientific computer installations they are of course adequate. Now let us look at the H.N.D. in maths, statistics and computing. This is a very good syllabus but I doubt that it is needed in addition to the City & Guilds 320 course or perhaps I should say that I believe that it will take the place of the City & Guilds 320. Here too the course is too mathematically biased for it to be of real use or real interest to commercial users.

No O.N.C. or O.N.D. exists in maths, statistics and computing as yet - I think such a course should be introduced (for the benefit of staff working with scientific users) but if it is introduced will it clash with the City & Guilds 319?

Now let us look at the B.Sc. There is a 3 - year full time B.Sc. in computer science and a 4 - year sandwich one. I prefer the sandwich one as I believe that computing is a practical subject which is best learnt on a sandwich basis. Here too the existing courses are too mathematically biased. All colleges do not wish for this and I know of three colleges which have tried to introduce computing courses with a fairly high commercial computing content but all have been prevented from doing so by the authorities who insisted that more mathematics should be written into the syllabi. More mathematics having been written into the syllabi, it was of course necessary to insist on an "A" level mathematics entry. This is a great pity because there are many young people who do not have "A" level mathematics who could make a wonderful contribution to computer work and who could reduce the acknowledged grave shortage of systems analysts. 90% of all computer applications are commercial rather than mathematical but education in computers in universities and colleges does not reflect this bias. In fact the bias is entirely the other way round and it is fair to say that 90% of computer courses are highly mathematical and very few courses exist which have a high commercial computing content. The perfect honours degree course should in my opinion be a four-year sandwich course with the first two years common to all students. I think the syllabus should then branch and those people who want to do highly mathematical and scientific computing could branch in one direction whilst those who wish to do commercial computing and data processing could

branch in the other. To my certain knowledge three colleges of technology would introduce these courses if they could obtain C.N.A.A. authority to do so.

If we look over to the commercial side you will see down at the bottom mention of a Royal Society of Arts course - this does not exist at the moment but I very much hope that the R.S.A. will introduce some form of certificate for punch and machine operators; it is also possible that they may introduce a certificate for junior programmers.

We now come to a gap; as you will see no O.N.D. or O.N.C. in commercial computing exists - I feel that an O.N.D. or O.N.C. would make a very satisfactory substitite for the 319 in so far as students working in commercial set-ups are concerned. I hope a move towards the introduction of such a course will soon be made.

Then we come to the H.N.D. and the H.N.C. An H.N.D. in commercial computing does not to my knowledge exist but an H.N.C. in business studies with commercial computing has been started at South West London College. This course consists of a six-month full-time block release period in the College followed by 18 months evening and directed private study. The H.N.C. is awarded after two years and after the students have passed a written examination and a viva and have completed a project and written an extended essay on a liberal subject. This is a great break-through and will I hope lead to a series of H.N.D., H.N.C., O.N.D. and O.N.C. courses in business studies with commercial computing - such courses would meet a very long felt need.

Wandsworth and Batersea College of Further Education have pioneered day-release-from-school-to-college courses so that several schools in the Putney area can send young women to college to learn about data processing one day a week during their last year at school. We started these courses because we felt that it was just as good (in fact a great deal better) to learn about data processing in college as it was to learn typing in school. These courses are two year courses - the first year on day release from school and the second, full-time in college.

Whilst on the subject of day release from school to college I would like to mention the running of the C ity & Guilds 319 course on a day release basis. We have found that this is not at all a good plan. Computing is such an urgent subject and the whole thing is so volatile that we have found that some young people doing this course on a day release basis have been unwilling to continue it because they say they have out-stripped the whole two-year syllabus within their first six months in their work sityation. I think, therefore, that if 319 is to succeed it will have to be run on an end - or block release basis. I emphasise the need for end-on courses because such courses are far more convenient for the employer and are therefore likely to be better supported.

I think that the best of these courses are the sandwich courses and I think that the Brighton type course is the answer to the shortage of systems analysts. After four years at Brighton (two of which have been spent in industrial training) the graduates should be good enough to become systems analysts six months after graduation.

If the students cannot obtain college-based or industry-sponsorship, these sandwich courses will fall into basest disrepute.

If one looks to the engineering side of any large firm one finds numerous, imaginative, well-endowed apprentice and university studentship schemes. On the commercial side one rarely finds anything similar. Surely in 1967 education for commerce, for management, for business should be as deep, as lengthy and as well endowed as education for medicine or engineering. I think a radical change in thinking is required here and I hope that in five years time commercial apprenticeships and studentships will be as common as engineering apprenticeships and studentships are today.

Apart from creating a continuing supply of educated personnel these schemes have other advantages - the turn-over rate for apprentices is usually very low - they get over the maximum "itchy feet" period when still in training. They put down roots and wastage falls.

There are financial advantages too - for each 52 week period you get £3.17.0d. per week per apprentice from the I.T.B. - paid, of course, in arrears (£413.0.0. in four years).

We are told that the U.K. will be short of 11,000 systems analysts in 1970.

This is what I call the grand design. It looks into the future quite a bit but this is what I believe the educational picture will look like. I think we will have many large multi-access real time computers in education. We will have anything up to 400 on-line stations on these big machines and I believe that they will be on-line to other colleges, to satellite machines, to schools, to colleges of education etc. etc.

The establishments (very probably the polytechnics) who own these machines will use them for internal administration, for research, for education in computing for all students (on both the arts and the mathematics side) and they will a so be used for computer aided instruction.

I think that these big machines will be sited in polytechnics because I think that although universities will have big machines themselves they will not in most cases have links outside because all their computing facilities will be required within the university campus to serve their own undergraduates, post graduates, research and administration etc.

I think there may also be a number of large installations doing educational administrative work i.e., the Universities Central Council on Admissions, the examining boards.

Computer Aided Instruction (CAI) is going to be a very very important thing and let nobody fear that the computer is going to take the place of the lecturer or the teacher. Far from it, C.A.I. will take a lot of the drudgery out of teaching and will leave the teacher free for more stimulating and creative tutorial work.

Remarks on AVERAGE B

1. The Flow-chart - This refers to a flow chart produced in the previous issue of Computer Education.

I am apalled that, after six hours instruction, students are unable to produce a flow-chart of a better standard than that shown (p.48, CEG Bulletin No. 00100).

Let us run through it, step-by-step.

a) Load and store control nos.

What are the control nos. and what do they control?

b) Load and store no.

Difficult to criticize. I might say that, in general, reading a number implies that it will be stored, so that "and store" could be considered redundant.

c) Last no?

How should I know? The last number pre cdes the - 999 I won't know until I've read the next number!

d) The NO on the left-hand exit from the decision box should be shown. Students should be encouraged to label all exits from flow-chart decision points. Continuing down the left-hand branch:

e) ADD. Add what to what? ADD is frequently used as an abbreviation for ADDRESS, but this seems unlikely here.

f) COUNT. Count what? or should I be performing an operation on a variable called COUNT?

g) (ritht-hand branch) Divide by N. What is N? It has not been previously defined, and the problem statement does not mention it. Even if I could find it, I wouldn't know what to divide by it.

h) Print out. Does this mean "print the characters OUT"? I think not. It probably means "print the result of the preceding calculation", but does not state this. PRINT RESULT would be much better. If the result has a name better still.

To quote the article on Flow-Charts (same issue, page 4), "the purpose of a flow-chart is to provide a clear description of how the problem is to solved." The flow-chart given certainly does not fit this description and, to a large extent this is the fault of the teacher.

2. The Program

Although written by novices, the coding given shows a lack of techniques which should be acquired early in a programmer's career.

a) The constants have to be read in from tape. This is wasteful of tape, time and key-punch effort. They should be set by program, where possible.

b) The final directive, (EXECUTE 100), causes execution to begin on a read instruction. Since the computer is faster than the hand, this implies that at least the first set of data will have to be on the same tape as the program.

This has several possible consequences:

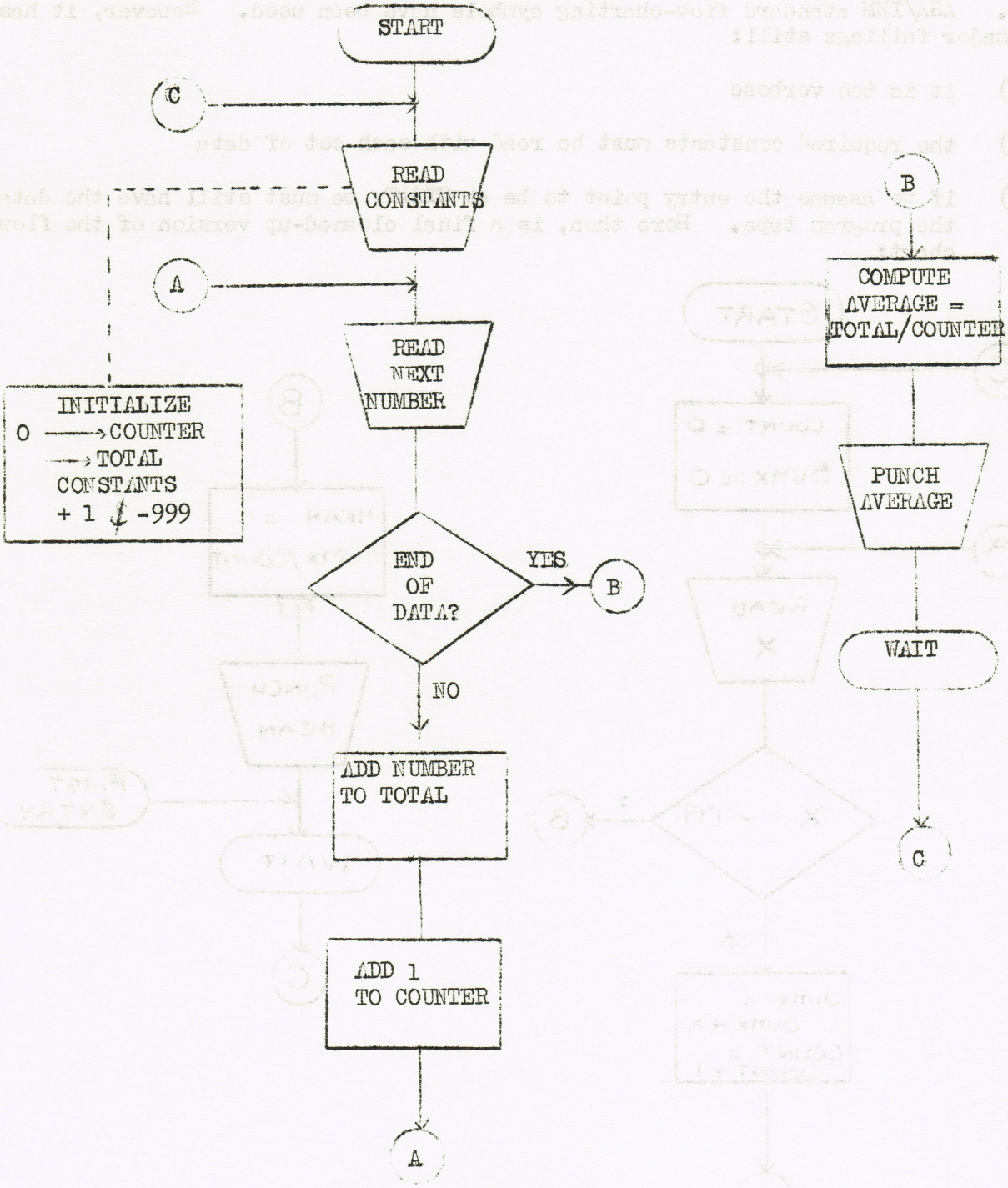
i) If all the data follow the program on the same tape, it is necessary that each data tape have the program punched at its head.

ii) If data are punched on a separate tape, then the program tape must be terminated by a set of dummy data or - 999. In both cases, an unwanted line of results will be obtained.

In writing a program, one should attempt to make its application general as far as possible. This means, in the simplest case, keeping program and data separate, and the easiest means of achieving this is by starting with a "wait" instruction (not to be confused with the C & G (WAIT) directive), which allows the operator time to insert a data tape in the reader. There is a misprint at location 121, where DIU 4 should read DIV 4. The redundant LDA at location 110 is a common occurrence in beginning programmers' coding, and is relatively unimportant if it does not occur too often, or in an inner loop.

The Flow-chart Again

Here is how the flow-chart might have been drawn:

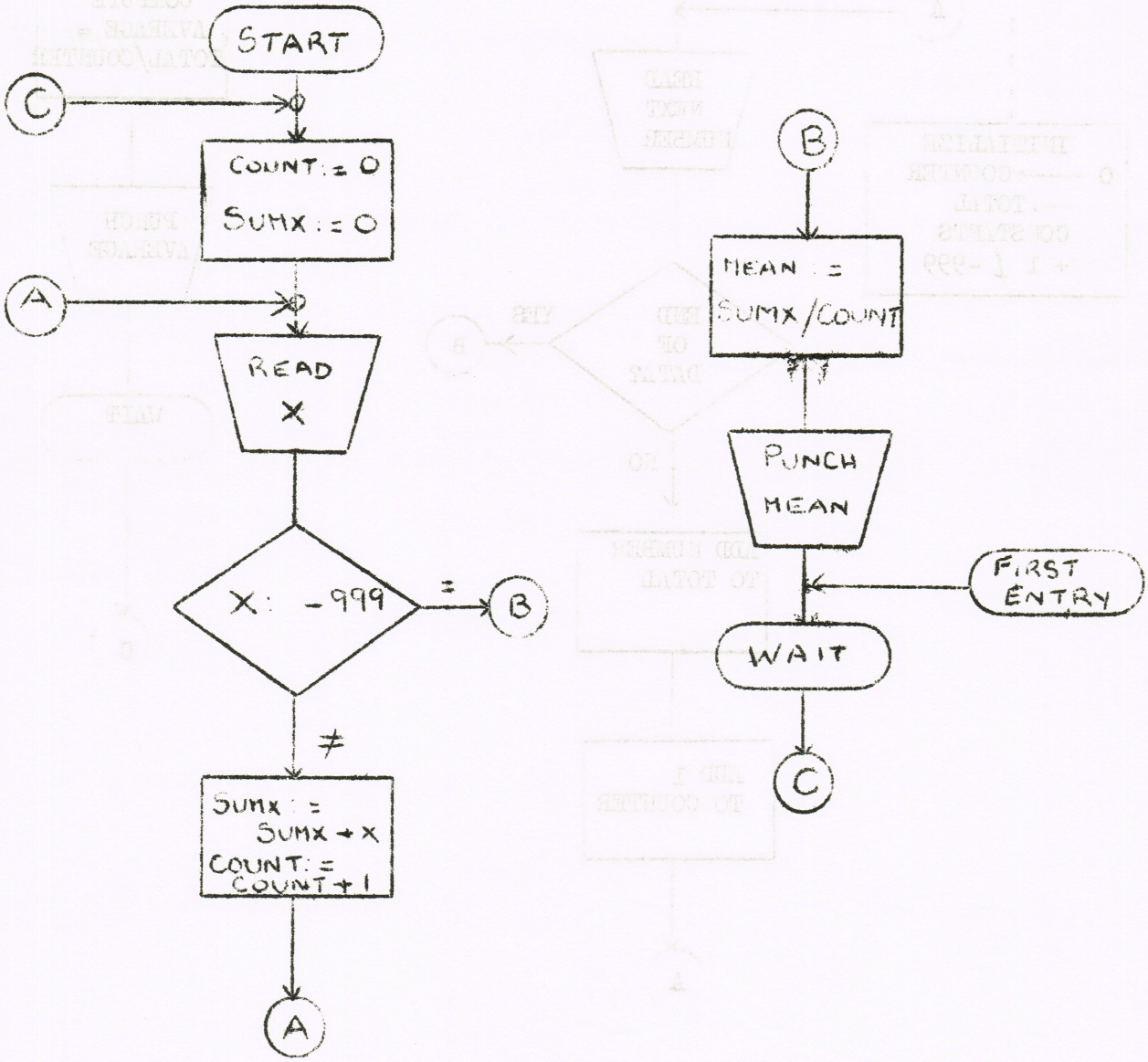

START
C
READ CONSTANTS
A
READ NEXT NUMBER
INITIALIZE
0 ⟶ COUNTER
⟶ TOTAL
CONSTANTS
+1 & -999
END OF DATA?
YES
B
NO
ADD NUMBER TO TOTAL
ADD 1 TO COUNTER
A
B
COMPUTE AVERAGE = TOTAL/COUNTER
PUNCH AVERAGE
WAIT
C

I hope readers will agree that this diagram is clearer than that originally published. ASA/IBM standard flow-charting symbols have been used. However, it has three major failings still:

a) it is too verbose

b) the required constants must be read with each set of data.

c) if we assume the entry point to be at START, we must still have the data on the program tape. Here then, is a final cleaned-up version of the flow-chart:



The following point should be noted:

a) This chart is machine-independent; that is, by using this chart, a programmer knowing the appropriate code could write the program for any digital computer. Thus,

b) It constitutes an algorithm

c) Standard symbols are used (ASA/IBM)

d) Use is made of ALGOL-like symbols (:=) in the processing boxes. If such symbols are learnt early in life, the transition to ALGOL is made simpler at a later date = or could be used.

e) The first entry is to a "wait", which allows a data tape to be placed in the reader.

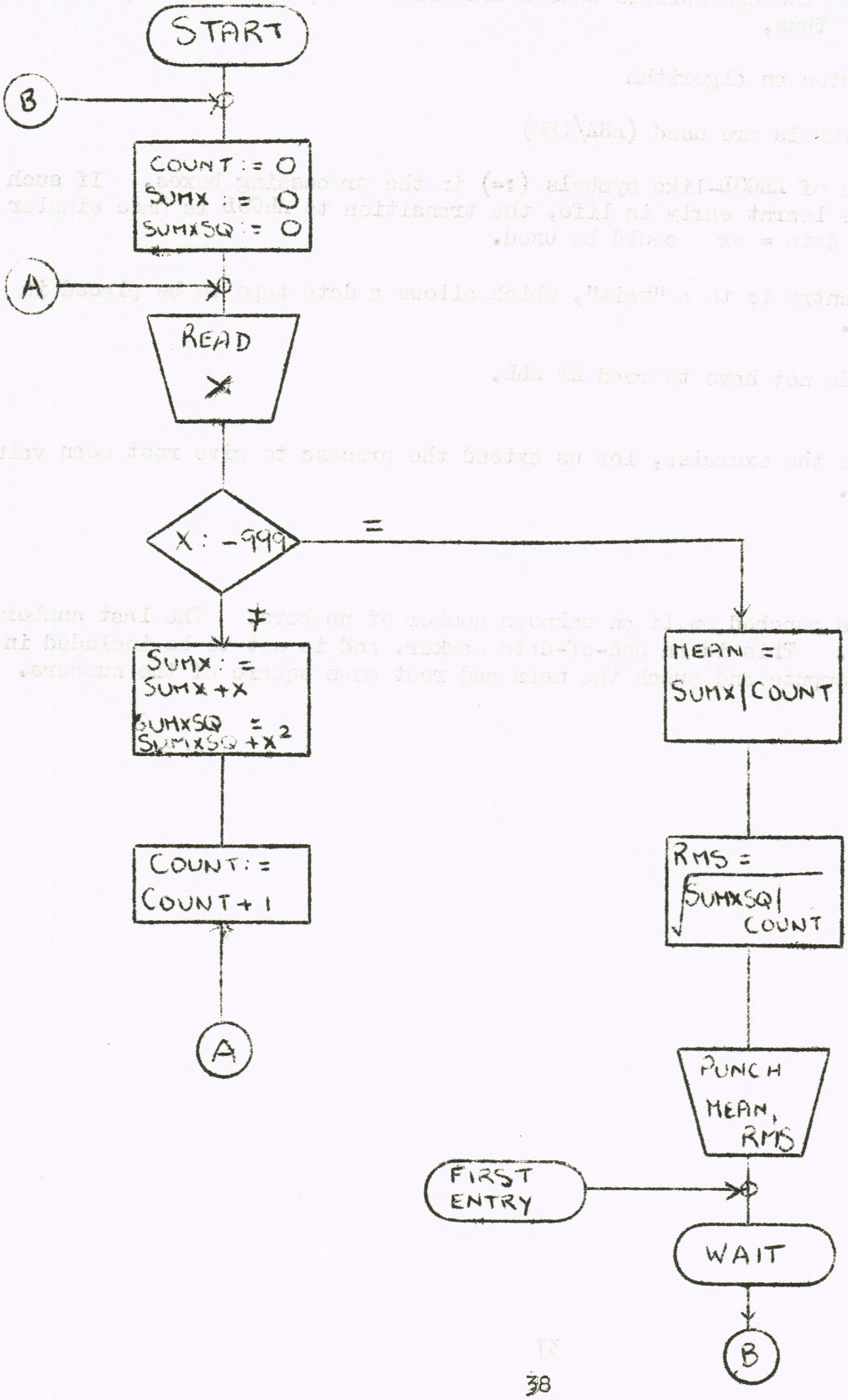
f) Constants do not have to read AT ALL.

Now, to complete the exercise, let us extend the process to give root mean values, in addition to means.

Problem Statement

A paper tape has punched on it an unknown number of numbers. The last number on the tape is - 999. This is an end-of-data marker, and is not to be included in the computation. Compute and punch the mean and root mean square of the numbers.

Flow-chart


START
B
COUNT := 0
SUMX := 0
SUMXSQ := 0
A
READ
X
X : -999
=
≠
SUMX := SUMX + X
SUMXSQ := SUMXSQ + X²
COUNT := COUNT + 1
A
MEAN = SUMX / COUNT
RMS = √(SUMXSQ / COUNT)
PUNCH MEAN, RMS
FIRST ENTRY
WAIT
B

CODING

```
     (TITLE) MEAN AND RMS
     (STORE 10)
 10 + 0              (see note 1)
 11 + 0
 12 + 0
 13 + 0
 14 + 0
 15 - 999            (see note 2)
     (STORE 100)
100  LDAN 0         (see note 3)
     STA 11
     STA 12
     STA 13
104  RNT 10
105  STA 10
     SUB 15         (see note 4)
     JEQ 119
     LDA 12
     ADD 10
110  STA 12 )
     LDA 10 )
     MLT 10 )
     ADD 13 )
     STA 13 )
115  LDA 11
     ADDN 1
     STA 11
     JUN 104
119  LDA 13 )
120  DIV II )
     SQT 130)       (see note 5)
     STA 14 )
123  LDA 12
     DIV 11
125  PNL
     PNT 5, 5
     LDA 14 )
     PNT 5, 5)
129  JST 100
130  PNL    )
131  JUN 129)
     (EXECUTE 129)
```

RMS Calculation & Output

STORAGE ALLOCATION

10	x
11	Count
12	Sumx
13	Sumxsq
14	RMS
15	- 999
100	PROGRAM

Notes on Coding

1. It is not necessary to place zeroes in the working area; in this case, however, it is immediately obvious that storage space has been reserved for constants and variables. Many systems require such reservations to be made.

2. - 999 is placed here, since the problem specifically states that this will terminate the input data. If we were to be completely general, a value could be read into this location.

3. LDA 0 might equally well have been used.

4. This is probably preferable to the (in this case) equivalent ADDN 999. The latter, should the end-of-data marker be changed, would also need changing.

5. This is a simple check that the computer is doing its job properly. A negative sum of squares indicates that an error has occurred. In this case a blank line is output, and a jump is made to the "wait" (JST) instruction. It was not felt necessary to include this in the flow-chart. An error message could be printed if we wished to be more sophisticated.

Having been away from the country for a short while, I have to some extent, lost touch with C & G mnemonic code. If, therefore, there are errors in my PNL and PNT instructions (designed to give an output format of xxxxx.xxxxx), I should be glad to know of them. In any case, these are frequently machine-dependent in their implementation. The original flow-chart says print but the coding produces output.

The author and readers, will, I hope, forgive my rather rough criticism of the original flow-chart. They should bear in mind that one of the most important tasks of the programmer is program documentation. If a programmer cannot communicate, he is of no use to his Company, and carried to an extreme, this makes him unemployable. It is essential that students should appreciate this, and that sloppy documentation should be corrected at a very early stage of their training.

Arthur S. Radford
Leicester College of Technology
Department of Mathematics
Computing Centre

JOINT CONFERENCE ON COMPUTER EDUCATION AND TRAINING

(28th October 1966 - Edinburgh)
Reviewed by N. E. Bowker (Enfield College)

This conference was sponsered jointly by the British Computer Society, the Institute of Chartered Accountants of Scotland, and the Scottish Council for Commercial, Administrative and Professional Education, and considered problems in relation to Scotland. The comments made, however, are almost entirely applicable to Britain in general.

Session 1 - WHAT IS AVAILABLE?

Mr. J. Shaw (Vice-chairman, Computer Education and Applied Research Study Team) surveyed the existing facilities in Scotland, and drew the following conclusions:-

1) Those who use computers in their work, but do not make a living by them, (e.g. mathematicians, engineers) are well catered for, within the context of their requirements.

2) Those who need to know about computers because their lives are affected by them are reasonably well provided for in "appreciation" courses, (although these are too often attended by those who should be receiving more detail).

Middle management are best covered, by exposure to courses organised by professional bodies and manufacturers, appreciation courses in their pre-employment training and exposure to their colleagues. The initiative for computer installations often comes from this level of industry!

Junior management are just beginning to be contacted via S.C.C.A.P.E. courses, but lower levels of staff are not dealt with at all at present, although schemes are under discussion.

A very serious omission is that top management are not covered at all..... and this is where decisions on the use or acquisition of computers are made.

3) Those who make their living by running computers are poorly provided for, the main provision being the City and Guilds course 319 and the Diploma in Systems Analysis and design, both courses running in Glasgow and Edinburgh.

	cycle time	training time
Operators	1 - 7 days	1 month
Supervisors	1 week - 3 months	6 - 12 months
Programmers	3 months - 1 year	6 months - 3 years
Systems Analysts	1 year upwards	2 years upwards (If ever?)

After emphasising the overlap between the grades, he discussed the problems in staffing caused by market demand. (He quoted an average service life of $2\frac{1}{2}$ years for a commercial programmer!) There was a lack of objective measures of ability, and the high salaries commanded in the present market made "casualties" very reluctant to transfer out of computing. Scientific and Commercial programmers had little in common.

The requirements of industry he divided into three categories, requiring different types of training.

Management: Managers who are not data processing specialists need training in the subject, so that they become competant users, not necessarily experts. Data processing is difficult to pick up casually, and present courses tend to be superficial, with too much emphasis on hardware.

Systems Analysts: This is the best defined area. The systems analyst must have a practical working background in industrial and commercial work, since his job is to formalise such practical knowledge. Training should include competance in programming, flow charting, system documentation, work measurement, and much practical experience with working installations.

Programmers: The use of high level languages is resulting the programmer becoming a "craft" employee, recruited direct from school. Training should probably be similar to that of laboratory technicians, with day release or evening classes leading to a recognised qualification such as City and Guilds.

Mr. P. Giles (Scottish Amicable Life Assurance Society) further developed the themes of "Professional" and "Technician" training.

The student Technician is likely to be weak in mathematics, so that numerical methods rather than theory should be taught. Since commercial work is concerned mainly with the handling of large quantities of data, mathematics is barely required at all!

The Systems Analyst on the other hand requires a carefully chosen width of training, as he must be able to improve on existing systems, not merely copy them. More emphasis on high-level languages is necessary in the present situation, with less time devoted to hardware.

Session 2 - WHAT IS WANTED?

Dr. A. Ashford (Head of management services, Jessop Savill Ltd.) analysed the needs of industrial users, first defining personel levels in terms of the "cycle time" before their work was checked by a superior, and gave an estimate of the training required by each grade.

Session 3 - WHO WILL DO WHAT?

Mr. W. A. Donaldson (University of Strathclyde) spoke of the two contradictory criticisms made by employers. On the one hand, that the products of the educational system know too little of the world, and required substantial further training, and on the other hand that the employer requires not a person trained in existing practices, but one who is capable of fundemental analysis. "Knowledge" or "Thinking power". In practice both processes require time, and in a course an allocation must be made to each.

One escape from the dilema is for Universities to concentrate on "Thinking", with courses in Operational Research and business studies such as Accountancy, whilst the Colleges of Further Education concentrate on "knowledge", with business studies again, but with more emphasis on the problems met by the Systems Analyst. Both types of course would have a core of computers and programming.

Such a division would exclude from O.R. many potentially effective people, and from systems analysis the graduates who now mainly occupy the field. This posed an important problem for the conference to discuss.

Mr. R.J. McQuaker (National Computer Centre) observed that his chief valve was as a listener, and then gave a detailed survey of sourses of training and potential trainees.

The sourses were

National Computing Centre	
Universities	
Colleges of Technology and Commerce	For Short
Industrial Management or Staff Colleges	courses
Commercial enterprises	
Computer Manufacturers	

Potential trainees are "non-professionals" such as line and top management, or "professionals" from computer managers down to operators.

The N.C.C. may provide training for top management, and supports training for other grades.

The "professionals" posed a larger problem, some 5000 Systems Designers per year being required for example, involving some 40 - 60 centres in the U.K. training 100 each. This corresponded to 600 -800 per year in Scotland. Operaters or programmers are trained mainly by the manufacturers and users.

In the discussion that followed, it was generally felt that Systems Analyst need not be graduates, given adequate training, and that a Scottish computing centre which was the equivalent of a teaching hospital was required.

Delegates attending the conference were given a questionaire, and the results of this are an interesting outcome of the event.

Programmers: Only moderate formal education was required, and 80% favoured technical colleges or manufacturer as a source of training, preferably by full-time intensive schemes. There was a lack of interest in evening classes, and the programmer - systems analyst career path was followed by 4% of programmers.

Systems Analysts: High educational requirements, moderate/extensive programming ability, with emphasis also placed on O.R. and business studies. Sources of training were spread evenly, with only manufacturers excluded. About 50% of analysts were graduates, and there was a large opinion in favour of post-employment, full time, training, and qualification by examination.

PLANNED CAREER DEVELOPMENT IN COMPUTERS

Phased Training Plan introduced by John Hoskyns and Company Limited

John Hoskyns and Company is introducing a sequence of interlinked training courses for companies wishing to establish comprehensive data processing educational facilities for their staff. These courses provide career development opportunities extending over several years for staff selected by their companies to attend.

Between courses students return to their parent companies to put what they have learnt into practice.

The courses are graded in five levels and will provide education ranging from an introduction to computers and computing, through basic programming and systems analysis, advanced programming and systems analysis, to project management.

In addition, a separate course had been specifically designed for higher management to enable them to control computer projects and to take an active part in formulating data processing policy. This course covers computer concepts, evaluation of alternatives (economics and risks), project organisation, planning and control.

This is believed to be the first comprehensive training scheme in the UK designed to produce practical professional data processing people for organisation using, or planning to use computers.

The phased training plan benefits both the individual and the company. The individual is taught by practising consultants and thus develops into a professional rather than remaining an amateur. By adopting such a comprehensive training plan, a company should be able to attract and hold its DP staff.

SPECIAL FEATURES OF PHASED TRAINING PLAN

One of the features of the Hoskyns Phased Training Plan is that each person can progress to the level appropriate to his ability, at a rate within his capability. It is anticipated that a person entering the scheme will normally attend one course a year.

The scheme is flexible. Provision is made for people with some data processing experience to enter the training plan at a level appropriate to their knowledge.

Although the training scheme is progressive and extends over several years, employers pay fees for each course separately. A report on the results attained by each student attending a course is given to the employer so that progress can be gauged.

Before being accepted for a particular course, each potential student must have reached a specified level of competence and knowledge. This will avoid one of the problems associated with many current computer training courses where the knowledge span if members grouped together on the same course is often too braod for many to receive any benefit from the training given - either because their initial knowkedge is too little, or too great.

Lectures will be given by consultants and invited speakers, and will be based on practical field experience.

Several weeks after each course a "feedback" seminar will be held to assess the effectiveness of the training given, the progress made by individual members in applying the knowledge gained, and to evaluate any criticisms or queries arising out of the course material. This, in conjunction with the use of working consultants as lecturers will ensure that the information content of every course is maintained up to date and that syllabi are altered to comform with the latest data processing de elopments.

Great emphasis will be placed on educating course members in computer concepts, applications and techniques. In other words, on the use and exploitation of computers rather than on details of hardware and software which, of necessity, tend to receive most attention in manufacturers' training courses.

PHASED TRAINING PLAN - brief details of course structure

Level 1 - INTRODUCTION TO COMPUTERS

Course aims To familiarise individuals with the logical thinking required to use computers.

- To provide a basic grounding in hardware and software.
- To illustrate by means of case studies the variety of ways in which computers can be used on commercial applications.

Duration - Two weeks

Level product - People ready for basic training in programming and in certain cases, systems analysis.

Level 2 - BASIC TRAINING

(i) Programming

Course aims To provide a thorough grounding in:

- Programming techniques as they apply to a particular computer system
- File organisation

- Operating systems
- Testing Methods
- Basic program strategies and standards

Duration - Basic programming three weeks
- Program strategy one week

(ii) Systems analysis

Course aims : To teach people systems study, design and implementation. These include;

- Project structure
- System concepts
- Basic commercial applications
- Systems standards
- Communications

Duration - Four weeks

Level product Individuals who:

(i) Are able to make an immediate and effective contribution to an installation's work in the fields of systems analysis or programming.
(ii) Have the proper foundation for quick development and later training in more advanced techniques.

Level 3 - ADVANCED TRAINING

(i) Programming

Course aims To prepare people experienced in programming for senior programmer positions. The advanced programming course will concentrate on:

- Programming team management
- Program planning and scheduling
- Requirements of software and utility programming
- Estimating
- Program audits

- Program and file library control
- Modification control

Duration - Two weeks

(ii) Systems analysis

Course aims To prepare analysts, experienced at a basic level, for senior analyst positions. The course will concentrate on:

- Systems team management
- Feasibility studies and economic evaluations
- Organisation concepts
- Design of integrated information systems
- Basic OR techniques

Duration - Two weeks

Level product Individuals who are capable of creating an effective and forceful systems or programming team fully equipped to develop and use the more advanced techniques provided on the course.

Level 4 - SPECIALIST TRAINING

Course aims Courses in this category will concentrate on providing experienced analysts and programmers with the background necessary to allow them to develop their own ideas in special application areas. Initially there will be three separate types of course:

- Real-time systems - design
- Real-time systems - programming
- Operational research techniques in computer applications.

Duration - Two weeks

Level product Specialists trained to deal with an organisation's particular system problems.

Level 5 - PROJECT MANAGEMENT

Course aims To provide course members with a thorough grounding in the work involved in managing a computer project. Specifically:

- DP organisation

- Staff selection, evaluation and appraisal
- Project control and scheduling
- Equipment evaluation and selection
- Site planning
- Management education

Duration - Two weeks

Level product Fully qualified DP managers who are acutely aware of the role of DP in an organisation and the need for proper economic justification of projects, sound planning and effective standards.

Level 6 -HIGHER MANAGEMENT

Course aims To highlight top managements' role in controlling computer projects from initial study to final implementation. The course will concentrate on:

- Computer concepts
- Evaluation of alternatives
- Project organisation
- Project planning and control

Duration - Three days

Level Product Senior executives with the ability to play an effective role in the budgeting, planning and control of computer projects.

Further details from:

John Hoskyns & Co. Ltd.,
Boundary House,
Furnival Street,
London, E.C.4.

Tel: 01-405-8693

Books & Articles - Short Reviews

Logic & Algorithms: with applications to the computer - R. R. Korfhage - J. Wiley

Chapter headings

1.	Set, Relations & Mappings.	2.	Boolean Algebras.
3.	The propositional calculus.	4.	A view of binary vectors.
5.	Algorithms and computing machines.	6.	The first order predicate calculus.
7.	Formal languages.	8.	A Brief History.

Introducing Computers - F. J. M. Laver - British Information Service - An Introductory text.

IFIP-ICC vocabulary of information processing - North Holland publ. Co. Holland, and American Standards Assoc. - Everyone should look at these books.

Proc. 3rd Annual Computer Personnel Research Conf. June 1965.

Project Prepart - S. A. Larsen

The IBM Systems Research Institute - an intra-company activity - F. S. Beckman.

A study of the effect of attitudes on the learning of computer programming - A. J. Bimonle.

Audio-Visual Aids for data processing systems and automation - D.P.M.A. International Headquarters, Park Ridge, Illinois.

Several articles in "The Evolving Society" Jan - Feb 1967.

An Electronic Cash and Credit System ($7.50)
American Management Association (New York) 1966 - An intelligent and convincing argument for the advent of the "Cashless Society" before 1984.

B.C.S. Computer Courses 1967. price 7/6

- A list compiled by the B.C.S. in association with the N.C.C.
- Better than previous years but courses are still of an extremely varying nature and tend to be described in misleading terms e.g. degree courses and short appreciation courses look alike.
