possibilities is much easier than to think of the answer unprompted.
3. There may be more than one right answer. Consider: A is a point on the tyre of a bicycle wheel; B a point halfway along a spoke. When the wheel is turned, the point moving more quickly is: A, B, neither (underline the right answer). Although A has the greater linear velocity, both have the same angular velocity.
4. With a problem, a simple arithmetical slip should be much less serious than an error in reasoning, and moreover, such an error may be minor or fundamental. Yet all the computer can tell is whether the student has underlined the right answer.
5. The computer cannot judge either discrimination of facts or development of a logical argument. This is the prerogative of the essay.
6. The teacher can be deprived of valuable feedback. All he knows is which questions were badly answered not why. This becomes apparent only on hand-marking an essay or a problem.

References

GROVES, P. D. (1968). Marking and evaluating class tests and examinations by computer, *The Computer Journal*, Vol. 10, No. 4, pp. 365–7.

HINCKLEY, C. C., and LAGOWSKI, J. J. (1966). A versatile computer-graded examination, *Journal of Chemical Education*, Vol. 43, p. 575.

SMITH, S. R., SCHOR, R., and DONOHUE, P. G. (1965). Computer grading of small numbers of laboratory unknowns, *Journal of Chemical Education*, Vol. 42, p. 224.

To the Editor
The Computer Journal

What is an analyst?

Sir,

Whilst Professor Peter Fellgett resorts to the distant and foreign past to suggest that an analyst is antonymous with catalyst (this *Journal* Vol. 12, p. 104), may I suggest that it is less far reaching to suggest that his respected title is antonymous with confessor.

As a confessor is one who confesses his sins, a Professor is presumably one who keeps them to himself.

Yours faithfully,
C. NEWBY

(Ex. pupil of Professor P. Fellgett, now a systems catalyst.)
IBM United Kingdom Limited
London EC2
25 February 1969

To the Editor
The Computer Journal

What is an analyst?

Sir,

Peter Fellgett argues for the University White Knights. I will argue for the business Red Knights.

A programmer per se can program only if he chooses or is given something to program. If he makes a mess of it, this soon becomes apparent during testing, and he has lost nothing save time and self-respect. Rarely are huge sums of money involved, or is the workload of thousands of people in a dynamic environment seriously affected.

In the environment of a large industrial or commercial system, a different situation exists. It is currently impossible for example to create an exact mathematical model of a large industrial enterprise. It is sometimes possible to do no more than create a simplified model of a part of it. Once the model is created, programming is relatively simple. Sir Paul Chambers has said for example, that economics may be a more difficult subject than nuclear physics. If results are any criteria, then on the available evidence this proposition could be held to be true.

The best results are achieved by computers in a business or industrial environment only if the business is examined from first principles. From such an examination major consequences flow. Many processes and functions which have become traditional are often found to be either fossilised or incompetent. Data Fields must be savagely reduced. Hence a new business system has to be devised, and the analysts who do these design studies have to understand the business to the extent that they can argue the facts on equal terms with the many managers who are affected, and to whom the concepts are often foreign. When at last a new system has been designed, managerial agreement to it must be reached *and subsequently maintained.*

It is these mutually agreed system documents which become the basis for subsequent programming. Sometimes system amendments have nevertheless to be generated by feedback of difficulties, diseconomies or errors in the system which become recognisable due to the detailed work of charting and programming, and the exact disciplines required. To minimise such system amendments, as well as for other obvious reasons, the analysts should at best themselves be programmers, and ideally they should be selected businessmen who, equipped with the required aptitudes, have had the necessary systems and programming training superimposed upon their business experience. It is the business and system knowledge and experience which in practice takes the longest time to acquire. Programming training and experience is more easily come by, and involves less time on the critical path.

In the Royal Army Ordnance Corps, we have few 'pure' programmers. All team leaders and Heads are first and foremost Army and Civil Service 'businessmen' with a great deal of superimposed computer experience and expertise. New entrants to the field are all given some 'business' training, where they lack it, and we attach great importance to this requirement. A programmer is of limited value until he has immersed himself in the system, in our environment at least. The system is a swimming pool, not a teacup. Our own studies involved well over 200 man years of systems design effort before we reached the point where some programming could start. Implementation of parts of our new proposals on a manual basis prior to delivery of our new large computer complex has already led to economies, which are a taste of things to come.

To say, as Fellgett does, 'that programmer remains the most honorific term' in my opinion and if the discussion is not merely semantic, bespeaks a limited vocabulary, and is naive and even perhaps dangerous, at least anywhere outside the microcosm of a Department of Applied Physical Sciences.

Yours faithfully,

R. L. ALLEN (Brigadier)

Inventory Systems Development Wing
RAOC
Didcot, Berks
28 February 1969

To the Editor
The Computer Journal

Matrix form for data dictionary

Sir,

King (1969) argues well the central need for a data dictionary in our too slowly evolving methodology for systems analysis and design. He also offers three basic attributes of such a dictionary. These are most helpful. May we explore his second attribute a little further—that of providing '. . . information on the grouping and structure of data'?

We agree completely with this requirement. However, a difficulty arises, since data is structured differently for different purposes. For example, 'Hours worked' within 'Man's Number' might be the input structure for a payroll. 'Man's Number' within 'Hours Worked' might be the output structure for statistics on overtime. King recognises this and proposes a four part dictionary of which part three provides a document for each major structure. Parts one and two, complete lists of data elements (or Master dictionary), are left unstructured.

An alternative approach can however be adopted which shows, in the Master dictionary, the underlying structure of the data—irrespective of its use on any particular occasion. This is done by constructing a data matrix which lists all the data items down the side. Along the top are shown their identifiers. Identifiers are defined as those items whose values are associated with single values of other items. For example, 'Date of Birth' may be the name of the complete set of birth dates for all persons on the payroll. 'Man's Number' is the identifier of Date of Birth since, given a value of Man's Number, we can identify a particular value of Date of Birth. There is a single Date of Birth for each Man's Number. Note that the reverse is not true.

Using a data matrix, King's dictionary can be expressed thus:

	Week	Day	Kiosk	Code
Week	x	1		
Amount per day		1	1	1
Banking per day		1	1	
Amount per week	1		1	1
Banking per week	1		1	

The five rows of the matrix can be interpreted in plain English like this:

1. Day identifies Week. Given a value of Day number, we can identify the Week it belongs to. The x in the first column indicates that Week identifies other sets, as well as having an identifier of its own.
2. Day, Kiosk and Code jointly identify Amount per day. Given a value for Day, Kiosk and Code, we can identify a particular value of Amount per day.
3. Day and Kiosk jointly identify Banking per day.
4. Week, Kiosk and Code jointly identify Amount per week.
5. Week and Kiosk jointly identify Banking per week.

Not only is this much shorter, but we can exploit identifier matrix when stating derivations. We can say, for example, that:

$$\text{Banking per day} = \Sigma \text{ Amount per day}$$

and, when we look at the matrix, it becomes clear that we have to add up all the values of Amount per day for a particular Kiosk/Day combination to obtain the Kiosk/Day's Banking per day. Similarly,

$$\text{Amount per Week} = \Sigma \text{ Amount per day}$$
$$\text{Banking per Week} = \Sigma \text{ Banking per day}$$

A fuller treatment of the method is given in a description of the dictionary as used in Systematics (Grindley and Stevens, 1968).

Yours faithfully,

C. B. B. GRINDLEY
W. G. R. STEVENS

Urwick, Orr and Partners Limited
Slough, Bucks
3 March 1969

References

KING, P. J. H. (1969). Systems Analysis Documentation: Computer-aided data dictionary definition, The *Computer Journal*, Vol. 12, pp. 6–9.

GRINDLEY, C. B. B., and STEVENS, W. G. R. (1968). Principles of the Identification of Information. Presented to FILE 68, an International Seminar on File Organisation held in Copenhagen, November 1968, sponsored by the IFIP Administrative Data Processing Group.

To the Editor
The Computer Journal

What is an analyst?

Sir,

Following the publication (this *Journal* Vol. 12, p. 104) of my previous letter under the above title, I have received a number of letters both *pro* and *con*, including one from an ex-student alleging that the negation of Professor should be Confessor (see above). Most correspondents appreciated that the light-hearted form of the letter disguised a serious purpose, and some further comment on this may be in order.

First, it is right to ridicule the assumption that tidy-minded classifications can ever encompass the multivariate capabilities of human beings, whose abilities will in any event soon appear pragmatically; by their fruits you shall know them. In general, the idea that real or useful results can be obtained by mere verbal classification has been described psychologically as pertaining to those who have never grown up from the stage in their development when they learnt to speak. In the particular application, the term Engineer can be applied to a chap in dungarees with a dirty great spanner, but is not scorned by those who have made it possible for members of the human race to see the Moon as a landscape and the Earth as a planet. Similarly, if the term Programmer was good enough for the academic who wrote a supervisor by himself between Christmas and Easter, may not anyone be proud of this title, from the girl who writes algebraic sub-routines upwards.

Secondly, I have been saddened by a particular kind of applicant for a university place who says he wants to be a 'systems analyst'. On interview, these applicants produce a series of statements which are initially difficult to compile, but by heuristic methods one gradually builds up in one's mind (and can verify empirically) a macrogenerator capable of translating into basic language, roughly as follows, 'I, the applicant, am not good enough at abstract thought to write useful programs, therefore I want to be in a position to tell other people to write them'.

It is sad indeed that this kind of misapprehension should exist, but it is hardly surprising. In schools the negation of comprehensive is apprehensive, and this is not for educational reasons (which may be valid or invalid, we do not yet have the evidence) but in fulfilment of political doctrines which confuse equality of opportunity with equality of achievement. Streaming is frowned upon, except when performance is deemed important as in competitive athletics. Specialisation has become a dirty word, in defiance of the reasoning that in as much as a specialist has learnt to do at least one thing well he is broader than the non-specialist, and of course even more so if he has specialised in more than one thing. From all this confusion, the sixth-form student receives again and again a macro which translates roughly as 'Do not bother to qualify yourself in anything, it will only get you labelled, and will put off those who have never learnt to operate at specialist level; aptitude falls like manna from heaven on the chosen few, and they are made Managers'.

These considerations are specific to computers in the following sense. In many practical activities, trained ability can enable a person to get further and faster by a factor of perhaps two or three, but in computing the factor may be 10^2 to 10^3. The average level of the approach to computers, weighted in proportion to total man-hours not (as is often done) attaching financially disproportionate importance to the academic peaks, has been far too low. Indeed the majority of computers are today not involved in producing new knowledge and wealth, but in non-productive accountancy tasks. Through no fault of the manufacturers, this market demand has distorted our thinking until computers have become so like accountants that they are now expected to be able to read black copy. With rare exceptions, our concept of the hardware structure of a computer has remained bogged down in the processor-store dichotomy of the early number-crunchers, yet this structure is inappropriate to the most important functions of computers in control, decision, and information handling, many of which are moreover not numerical at all.

If ever there was a case for SET, it is in computers. Those employed non-productively should attract tax, and those employed for useful production should attract a grant. Then there might be some good computers designed, having deep and far-reaching innovations.

Yours faithfully,

PETER FELLGETT

Department of Applied Physical Sciences
Whiteknights, Reading
18 April 1969

To the Editor
The Computer Journal

Sir,

We are engaged in a project to acquaint secondary school teachers of possible computer application in the instruction and administration of classes in English literature and composition. We would be very grateful to any of your readers who might provide us with some practical applications that could be used as examples in a computer orientation program. It goes without saying that any contribution will be fully acknowledged and that we will share our information freely.

Yours faithfully,

MARTIN J. BIRNBAUM

Teaching Research
Monmouth, Oregon 97361
U.S.A.
8 July 1969

To the Editor
The Computer Journal

Sir,

Concerning my paper: 'Error estimates for Runge–Kutta type solutions to systems of ordinary differential equations', published in your May 1969 issue, the following erratum has been brought to my attention by Peter Basnett of the Electricity Council Research Centre.

In the last equation of process (8), the coefficient of k_6 should be −14756/1392300 and not −14759/1392300 as printed.

Yours faithfully,

R. ENGLAND

7 Barns Hay
Old Marston
Oxford
16 June 1969