

MARKSCHEME

November 2001

COMPUTER SCIENCE

Standard Level

Paper 2

1. (a)

SMONTH	P	MONTHS (P)	SMONTH # MONTHS (P)
"March"	0	"January"	true
"March"	1	"February"	true
"March"	2	"March"	false

Allocate [2 marks] for each correct line (obviously the last two lines only as the rest is given in the question). One mistake in a line gives [1 mark] and two mistakes gives zero.

(b) return P+1 instead of P [2 marks]

(c) One possibility is as follows:

```
function TOTAL(val SMONTH string val NDAYS integer)
  result integer
  declare I, N, P integer
  N<-- MONTHNUMBER(SMONTH)
  if N>12 or NDAYS>DAYS(N) then return -1
  else P<-- 0
    for I= 1 upto N-1 do
      P<-- P+ DAYS(I)
    endfor
  P<-- P+NDAYS
endif
return P
endfunction TOTAL
```

Algorithms may vary but marks can be allocated in the following way:

pass by value of SMONTH and NDAYS [1 mark]

declaration of variables used in function [1 mark]

correct call to MONTHNUMBER [1 mark]

check on validation of result [1 mark] and NDAYS [1 mark] return -1 if neither valid [1 mark]

putting P to zero before the loop [1 mark]

correct pass through DAYS up to N-1 [2 marks] accumulating number of days [1 mark]

add on NDAYS [1 mark]

correct return [1 mark]

```
(d) declare FIRST, SMALLPOS, CURRENT integer TEMP string
for FIRST = 1 upto 11 do
  SMALLPOS<-- FIRST
  CURRENT<-- SMALLPOS+1
  while CURRENT#12 do
    if MONTHS (CURRENT) <MONTHS (SMALLPOS)
      then SMALLPOS< -- CURRENT
        CURRENT< -- CURRENT+1
      endif
    endwhile
  TEMP<-- MONTHS (FIRST)
  MONTHS (FIRST)<-- MONTHS (SMALLPOS)
  MONTHS (SMALLPOS)<-- TEMP
endfor
```

Looping may be implemented differently but essentially points should be allocated as follows:

Loop from 1 to 11	<i>[1 mark]</i>
allocate SMALLPOS and CURRENT	<i>[2 marks]</i>
loop until the end of array	<i>[1 mark]</i>
correct test for MONTHS (CURRENT) <MONTHS (SMALLPOS)	<i>[1 mark]</i>
place CURRENT in SMALLPOS	<i>[1 mark]</i>
increment CURRENT	<i>[1 mark]</i>
correct end of loop	<i>[1 mark]</i>
swap MONTHS (SMALLPOS) with MONTHS (FIRST)	<i>[2 marks]</i>
correct end of loop	<i>[2 marks]</i>

2. (a) very small change in one element (for example wind) can have effect on all other elements **[2 marks]**
 small change in elements can affect the time and place that weather will arrive **[2 marks]**
 the equations have to be continually fed with these variables to give latest changes and cannot predict what next change will be **[2 marks]**
 prediction equations are too simple and use too few variables to hold for more than a short period **[2 marks]**

Any of the above and many more are acceptable. Give [1 mark] for reason and [1 mark] for explanation.

- (b) short term uses fewer variables and restricted number of equations **[1 mark]** whereas long term includes many complex equations and many more variables **[1 mark]**
 short term is repeated on measurements as they are sent now **[1 mark]** long term uses archived data and statistics **[1 mark]**
- (c) One advantage **[1 mark]** justified **[1 mark]** and one disadvantage **[1 mark]** justified **[1 mark]**.

For example, an advantage is that changing secondary storage would mean:

- more storage
- taking less space
- easier to mount/dismount
- quicker to access

Disadvantages:

- not compatible with archived data so far
- means rewriting all the old data onto new medium
- could mean changing software

- (d) sending data from collection centre to central computer **[1 mark]** or to regional television company **[max 1 mark]**
- (e) **[1 mark] for each of the following points [max 4 marks]**
- set up equations to accept data
 - feed in current variables
 - compare forecast with what actually happens
 - repeat over a period time that covers a wide range of seasons, temperature, humidity *etc.*
 - take archive data
 - from a wide range of conditions
 - feed variables into model
 - compare forecast with the known resultant climate

- (f) *Accept any of the following characteristics [1 mark] with reason for which suitable [1 mark]. Do not give more than [4 marks] for characteristics or [4 marks] for reason. [max 8 marks]*
- parallel processing or pipelining needed to perform millions of calculations
 - precision arithmetic since calculations complex and small errors could mean different forecast
 - large immediate access store (accept RAM) needed for the number of variables and results that have to be processed
 - fast processing speed since each calculation is complex
 - front end processor to deal with input and output of enormous number of variables and results
- (g) *Give [1 mark] for correct online and [1 mark] for correct batch process.*
- On line process:
- would be the acceptance of data from the collection points
- Batch process could be:
- sending the current forecast after manipulation to the television centres at a set time each day
 - archiving data at a set time each day/week
- (h) *[1 mark] for any of the following:*
- no need to since data not secret
 - does not mean anything until put together with the rest of the data
 - saves time in sending and receiving
 - too much data to slow down with encrypting/decrypting
- (i) Data sent to regional television centres *[1 mark]* then resent to their own analysts *[1 mark]*.

3. (a) (i) A single task process is one where the computer is running one program at a time.
[1 mark]
- (ii) collecting and sending of data from the instruments *[1 mark]*
- (b) text to be read out *[1 mark]* by printer *[1 mark]*
pictorial output imposed on map of region *[1 mark]* video/screen output *[1 mark]*
[max 4 marks]
- (c) data input from extra instruments *[1 mark]* from data vendors *[1 mark]*
processing *[1 mark]* labelled *[1 mark]*
output to printer *[1 mark]* output to screen *[1 mark]* *[max 6 marks]*
- (d) Give *[1 mark]* for specific example and *[2 marks]* for description.

Multi-tasking between processing the weather data, word processing a document, creating DTP document

[1 mark] for giving two of the above processes that could be taking place

[1 mark] for any of the following *[max 2 marks]*

- system performs one process for a fixed time or until interrupt
- stores relevant data and addresses
- moves to second process and repeats
- user unaware that there has been a swap since rapid compared to user

Example may be included in description or stated separately.
