THE BCS PROFESSIONAL EXAMINATIONS
Certificate

April 2006

Examiners' Report

Software Development

Question 1

1. A doctor's patient records are stored for each complete year in a sequential file <patrec05>. Each patient record contains the following items:

NHS number	(4 digits)
Name	(20 characters)
Category	(one from 'child', 'teenager, 'adult', 'pensioner')
Postcode of address	(8 characters)
Repeat medication needed	(Yes / No)
Number of visits to surgery	(integer)
Major illnesses code	('A' or 'B')
Description	(20 characters)

a) Write an appropriate description for these records in a suitable programming language. State which language you are using. **(5 marks)**

b) A program is required to print a report which lists details of only the patients who live in a particular part of the city, which is identified by the first four characters of the postcode, e.g. 'TS12', which is requested interactively. The printed details are given under the headings below; that is,

NHS Number	Name	Surgery Visits
1200	JOE BLOGGS	32

At the end of the report the total number of patients for that part of the city is printed as well as the total number of patient records on the file. The ratio of these is also printed as a percentage.

Write an algorithm for this program in pseudocode or structured English. **(15 marks)**

c) Develop a program for the report as described in *b).* Use the record description from part *a)* but do not copy it out again and indicate where it belongs in your program code. **(10 marks)**

Answer Pointers

a)

```
TYPE   longstring  = PACKED ARRAY[1..20] OF CHAR;
        shortstring = PACKED ARRAY[1..8] OF CHAR;
       Posnum = 1..50;
       patient = RECORD
       NHSnum:INTEGER;
       name : longstring;
       category:(child, teenager,adult,pensioner);
       Repeatmed:BOOLEAN;
       Postcode:shortstring;
       Visits:Posnum;
```

```
        code:CHAR;
        majorill:longstring;
        END;
```

Examiner's Guidance Notes

a) Quite popular question with the full range of marks awarded. VERY few stated what language they were using. For about 30% of the candidates part (a) was all that they answered, restricting themselves to a maximum of 5 marks. Marks were particularly awarded for each instance of a different and correct type, for which there are alternatives.

Answer Pointers

b)

First algorithm

```
Initialise
OPEN file <patrec05>
Write report caption
WHILE NOT end-of-file <patrec05>  DO
        READ next record from file <patrec05>
        Process one patient record
ENDWHILE
PRINT  report footer
CLOSE file <patrec05>
END
```

b) Developed Algorithm

```
PRINT title
PRINT “input first 4 letters of postcode”
INPUT  district
ASSIGN <patrec05> as suitable data file
total = 0
patct = 0
{write report caption}
PRINT “NHS number  name  visits”
WHILE NOT EOF(patrec05)  DO
        BEGIN
        READ one record from <patrec05>
        INCREMENT  total
{process one patient record}
        IF record.Postcode = district THEN
                BEGIN
                INCREMENT  patct
                PRINT  NHSnum   name   visits
                ENDIF
ENDWHILE
{write report footer}
PRINT “total patients =”  total   “from wanted district =”  patct
average = 100.0 * patct / total
PRINT “average =”  average
CLOSE file <patrec05>
END
```

Examiner's Guidance Notes

b)

Inappropriate declarations were made here such as void main()   #include <iostream.h> (needed to run programs in C.) Some devoted space to variable declarations as well. Such declarations are NOT made in an algorithm unless type is central to solving the problem. Some gave a program here and copied it out again to answer part c).

Few gave any development of the original algorithm where it was needed

e.g. 'print details'→ WRITE(NHSnum, etc)

c) Develop a program for the report as described in b). Use the record description from part a) but do not copy it out again; indicate where it belongs in your program code.

(10 marks)

Answer Pointers

c)

Complete program in PASCAL (can be omitted from report if necessary)

```
PROGRAM doctor(INPUT,OUTPUT,patrec05);
{Program reports on patients in particular city district}
{Declaration of TYPE and RECORD belong here}
VAR patrec05:FILE OF patient; onerec:patient;
    total,patct,pos:Posnum;  district:shortstring;
    average:REAL; ch:CHAR;
BEGIN
WRITELN('patient visit report 2005 for particular district');
WRITELN('Input first 4 letters of postcode');
REPEAT  READ(ch)  UNTIL ch in ['A'..'Z'];
district[1] := ch;
FOR pos := 2 TO 4 DO READ(district[pos]);READLN;
FOR pos :=5 TO 8 DO district[pos] := ' ';
WRITELN;
ASSIGN(patrec05, 'C:\PROPAS\SOURCES\patrec05');
RESET(patrec05);
total := 0;  patct := 0;
{write report caption}
WRITELN('NHS number',' ':10,'name',' ':16,'visits');
  WHILE NOT EOF(patrec05) DO
     BEGIN
     READ(patrec05,onerec);
     FOR pos := 5 TO 8 DO onerec.Postcode[pos] := ' ';
     total := total + 1;
     {process one patient record}
     IF onerec.Postcode = district THEN
               BEGIN
               patct := patct + 1;
               WITH onerec DO WRITELN(NHSnum, ' ':7,name,'
':3,Visits:2)
               END
          END;
{Write report footer}
```

```
IF patct > 0 THEN average := 100.0*patct / total  ELSE average :=
0.0;
WRITELN('total patients = ',total:4,'  from wanted district =
',patct:4);
WRITELN('RATIO  = ',average:6:3,'  percent');
CLOSE (patrec05,FALSE);
END.
```

Examiner's Guidance Notes

c)

Marks given for added value in the program such as appropriate variable declarations, file declaration for <patrec05>, detail in output statements. Copying the algorithm out unchanged. Did not gain any marks.

Question 2

2. *a)* Dry run the algorithm below using the values a =16, d=3 and N=5. **(15 marks)**

```
Line number     Code
1               READ(a,d,N);
2               prod :=1;
3               a2j := a;
4               WHILE d > 0 DO
                    BEGIN
5                   IF odd(d) THEN prod := prod*a2j MOD N;
6                   d := d DIV 2;
7                   a2j := SQR(a2j) MOD N
                END;
8               WRITELN('result = ',prod:5)
9               END.
```

b) Write BRIEF notes on the use of variable identifiers and comments in the algorithm in part *a).* **(4 marks)**

c) Re-write the algorithm as an integer FUNCTION called Fermat with three parameters representing a, d and N. Incorporate in your re-written algorithm any improvements you noted in your answer to part *b).* **(11 marks)**

Answer Pointers

a)

It is necessary to identify those variables which change during the dry run. Thus (a2j, d) obviously gets altered while (N, a) do not.

It is also useful to have a column for the conditional expressions WHILE and IF, indicating whether the controlling condition is true or not.

The dry run starts with a = 16 d = 3 and N = 5 as specified

Line no.	Instruction	a2j	prod	d	WHILE	IF	Other/output
1	READ	?	?	3	?	?	A=16 N=5
2	Assign	?	1	3	?	?	
3	Assign	16	1	3	?	?	
4	WHILE	16	1	3	true	?	
5	IF	16	1	3		true	1*16 MOD 5 = 1
6	Assign	16	1	1			
7	Assign	1					16*16 MOD 5 = 1
4	WHILE				true		

5	IF					True	1*1 MOD 5 = 1
6	Assign			0			
7	assign						1*1 MOD 5 = 1
4	WHILE				false		Loop ends
8	WRITE						result = 1

Examiner's Guidance Notes

a)

Generally this question was done well. Nearly all candidates have adopted the tabular method for dry runs. However it saves time if the identifiers which change value during the run are identified and given columns, while those that remain constant (e.g. 'a') do not. Some candidates did not understand the 'DIV' and MOD' operators and gave decimal values for results. Others did not provide columns for the logical results, but it is satisfactory to indicate these in a general-purpose column the right. Some indication of the instruction must be given on each row, either by line number or part of the instruction. Both are not needed but this was not penalized – it makes the answer easier to mark!

Answer Pointers

b)

1) Poor choice of variable names. NONE of these suggest what they contain, particularly 'a' 'd' 'N'

2) No comments in the code nor statement what it does as a whole.

It is essential to make clear what names in the code have been altered, preferably using a table as below:

Old name	a	D	N	a2j	Prod
New name	astart	Divisor	Nrem	Apower	Product

Examiner's Guidance Notes

b)

Some candidates did not understand what was wanted here. Others put down nothing about 'comments' despite this being asked for. Others wrote lengthy descriptive material about good choice of names but did not apply it to the given question.

Answer Pointers

c)

```
FUNCTION Fermat(astart, divisor, Nrem:INTEGER):INTEGER;
VAR apower,product : INTEGER;                    (3 Marks)
BEGIN
product := 1;
apower := astart;
WHILE divisor > 0 DO
     BEGIN
     IF ODD(divisor) THEN product := product * apower MOD Nrem;
     divisor := divisor DIV 2;
     Apower := SQR(apower) MOD Nrem;
     END;
Fermat := product
END:  {Fermat Function}                          (8 Marks)
```

Examiner’s Guidance Notes

c)

Marks were given for ‘added value’ like meaningful names for identifiers (particularly ‘a’ and ‘a2j’). Any improved identifier names quoted in (b) should have been used here. A FUNCTION declaration was required with appropriate types assigned to variables. One or two comments were also expected, particularly one giving the purpose of the code. It was particularly expected that the function had parameters (a, d, N) which brought in the values, so a ‘READ’ statement was NOT needed in the function. Marks were NOT given for merely copying out the code unchanged.

Question 3

3. *a)* The array DATA has entries from index 20 to index 80 which are sorted in ascending order. The task is to find the index *pos*ition of the first *rep*eated number which is above a certain chosen value. Write a function called *pos_rep* in a programming language of your choice with a parameter *min* which will contain the lower limit. The function should return the number zero if no qualifying value in DATA can be found.

For example, when the DATA array contains the numbers as below then *pos_rep*(10) would have result 25 and *pos_rep*(2) would have result 21.

DATA											
Index	20	21	22	23	24	25	26	27	28	29	----
Value	1	3	3	7	8	14	14	15	---		

(20 marks)

b) Trace the call of your coding of *pos_rep* with parameter 10 and with the array DATA initialised as in part *a).*

(10 marks)

Answer Pointers

(a) Expect answers similar to

```
int pos_rep(int min){
      for(int i=20; i<80; i++){
            if(DATA[i]==DATA[i+1] && DATA[i]>min)
                  return(i);
      return(0);
}
```

or

```
function pos_rep(min:integer):integer;
ver i, posn:integer;
begin
      i:=20; posn:=0;
      while i<80 and posn=0 do begin
            if DATA[i]=DATA[i+1] and DATA[i]>min
            then posn:=i
      end;
      pos_rep:=posn
end;
```

There is an alternative strategy which is:
i) to use a loop to find the first index m of the array where DATA[m]>min
ii) to use a second loop starting at m to find adjacent equal values

(b) trace with min=10

A	B	C	D	E	F	G	H	
I	DATA[i]	B==D	DATA[i+1]	C&&G	DATA[i]	F>H	min	
20	1	False	3	False	1	False	10	

21	3	True	3	False	3	False	10
22	3	False	7	False	3	False	10
23	7	False	8	False	7	False	10
24	8	False	14	False	8	False	10
25	14	True	14	True	14	True	10

Value of i=25 returned

Examiner's Guidance Notes

(a) Some candidates apparently missed the significance of the statement in the question that the array was sorted in ascending order. This typically meant that their programs were more complicated than they needed to be with an extra loop in them.

Many candidates in this and other questions seemed anxious to include input (*read*, *scanf*, *cin*) and output (*write*, *printf*, *cout*) statements into their code. For example in this answer they might input the value of *min* inside the function *pos_rep* thereby missing the point completely of having *min* arrive as a parameter.

Some candidates were unwilling to work with subscripts 20 to 80 of the array and decided to change to 0 to 60. Presumably they believed (wrongly) that for loops have to start counting at zero.

(b) The invitation to 'trace' a program should always be taken to mean drawing up some kind of tabular presentation rather than writing in essay style.

There were cases where the trace bore little or no resemblance to the program given in part (a). To gain good marks the trace had to be a trace of the actual program given in part (a) quoting the correct variable names and the actual steps that the program would take.

Question 4

4. *a)* Choose either version A (written in C) or version B (written in Pascal) which perform identical operations and then provide comments suitable for the following lines of code:

i) line 1
ii) line 3
iii) line 4
iv) line 6
v) lines 8-10

(20 marks)

	Version A	Version B
1	void a(int b, c)	procedure a(var b, c : integer);
2	{	begin
3	int d, e;	var d, e : integer
4	for(d=b; d<c; d++)	for d:=b step 1 to c-1 do
5	{	begin
6	if(v[d]>v[d+1])	if v[d]>v[d+1]
7	{	then begin
8	e=v[d+1];	e:=v[d+1];
9	v[d+1]=v[d];	v[d+1]:=v[d];
10	v[d]=e;	v[d]:=e
11	}	end
12	}	end
13	}	end

b) It is desired to further modularise this code by making lines 8-10 into a separate subprogram (function or procedure). Write out the subprogram and rewrite lines 8-10 to make use of your subprogram. **(10 marks)**

Answer Pointers

(a)

(i) purpose of function *a* is to bubble the largest element of array *v* in the subscript range *b* to *c* leaving it in index *c.*
(ii) local variable *d* is a counter for the loop, *e* is a temporary variable to handle a swap
(iii) loop is to visit every subscript of *v* in the range from *b* up to *c-1*
(iv) if a larger element is found at index *i* adjacent to a smaller one at index *i+1*...
(v) ... then swap the values of the adjacent elements

(b)
change lines 8-10 to

```
swap(d,d+1);
```

where function swap defined as

```
void swap(int a,b){
        int t=v[a]; v[a]=v[b]; v[b]=t;
}
```

[or equivalent in other languages]

N.B. There is a much more sophisticated answer possible using variable parameters (in Pascal) or * and & (in C)

Examiner's Guidance Notes

(a) The phrase *provide comments* in the question was meant to be interpreted as an invitation to read and understand the program and then construct the kind of comment that the original programmer might have written. Suppose the following line was being considered

```
c=2*pi*r;
```

There is a huge difference between writing

Two is multiplied by pi and then by r and the answer stored in c

and

The circumference (c) of the circle is calculated from its radius (r)
The first is a very literal description of what will happen, the second is a much higher level of understanding and the language used is related to the problem at hand. In this question it was hoped that candidates would recognise that the code was bubbling the largest value to the top of the array by swapping adjacent values where necessary. Using the vocabulary *bubble*, *swap*, *adjacent*, or similar would indicate recognition of what the program was doing. [Note however that this program does not go as far as *sorting* the array as some answers claimed]

So for line 3 which in the C version is "int d,e;", the comments (i)-(iv) below are worth progressively more marks

(i) variables d,e are declared - 1 mark
(ii) variables d,e *of type integer* are declared - 2 marks
(iii) *local* variables d,e of type integer are declared - 3 marks
(iv) d is a local variable of type integer *used as a for loop counter*;
e is a local variable of type integer *used as a temporary variable during a swap* - 4 marks

Note that it was incorrect to refer to b, c, d or e as *global* variables, but it would be acceptable to refer to the array v as global.

Note that the question did not ask for *mistakes* in the code to be found and described.

Some credit was given to answers which merely explained the effect of the code without using the bubble, swap, adjacent vocabulary, however it is never acceptable to describe the *assignment* v=e by using the phrase *v equals e*. Candidates must use an action word like assigned (the value of e is assigned to v) or becomes (the value of v becomes e).

Similarly it is crucial to distinguish between the value of an array at index i and the value of i. Far too many candidates in trying to describe their understanding of if(v[d]>v[d+1]) wrote *if the value of d is greater than the value of d+1* when they should have written *if the value of the array v at index d is greater than the value of the array v at index d+1*.

Several candidates confused the *declaration* of a variable with the *initialisation* of a variable. Thus

int v; can be described as declaring integer variable v
v=0; can be described as initialising v to 0
int v=0; can be described as declaring v and initialising it to 0
int v; *cannot* be described as initialising a variable

Several candidates tried to explain a for loop (line 6) by wrongly using the word *until* when the word *while* would have been correct. So in the loop for(i=0;i<10;i++){...} the loop can be said to

continue *while the value of i is less than 10*
or
continue *until the value of i is greater than or equal to 10*

(b) This part was answered quite poorly. Several answers completely omitted this part thereby missing out on 10 marks. Of the candidates who did answer this part there were many which omitted the 'call' of the new subprogram and many with inappropriate formal or actual parameters.

Several answers had subscripts in formal parameter lists. When declaring a function e.g. void f(int p1,p2){...} and the formal parameters (p1 and p2 in this case) are being declared, there will never be any subscripts. This means that void f(int p1[i],p2[j]){...} must be wrong.

Section B

Question 5

5. Write a program to merge two distinct lists into a single linked list. Both of the initial lists contain an integer and the pointer and are both in ascending integer order. This order must be preserved in the merged list. The lists have the same length. They are set up by the procedure *setuplist(head)* which returns a pointer to the head of the list. You <u>must not</u> write this procedure but indicate where appropriate calls to it would be made in your code.

(12 marks)

Answer Pointers

```
Algorithm / pseudocode
Variable Declarations
Code for setuplist(head)
BEGIN {top level}
Setuplist(head1)
Setuplist(head2)
first ← head1;   second ← head2;
head3 ← NIL;
WHILE ((first <> NIL) AND (second <> NIL)) DO
   BEGIN
   NEW(third);
   third^.next ← head3;
   IF first list data item >= second list data item THEN
Copy FIRST list member into new list
      ELSE
Copy SECOND list member into new list
   head3 ← third
   END
```

Examiner’s Guidance Notes

This is an area of the syllabus where students memorise coded linked-list programs for setting up a list, inserting a new member and the like. Nobody had seen this particular problem before, so the memory-folk were confounded. A small number did get it right by beginning with a diagram showing what was needed with the two lists already set up (Some wrote out setting-up code, despite being asked not to do so.) Others launched straight into code with statements like *WHILE list^.ptr <> NIL AND found = FALSE*
without any idea what they were trying to do.
If adequate time is not available for the students to discover how to build or modify lists using pointer diagrams, they had best leave this subject alone. The Examiner considers learning the code to be unprofitable.

Question 6

6. Write a program which takes as input any 4-digit year and which prints out who was then President of the United States. Should the year precede 1933 or fall after 2005 an appropriate error message should be output. The necessary data e.g. **Clinton, 1993, 2001** may be stored on a previously written file or in DATA statements.

(12 marks)

Complete Program in QBASIC

```
PRINT "Presidents of the UNITED STATES of AMERICA"
DIM name$(10)
PRINT "input request year"
INPUT inyr
ct = 0
READ name$, startyr, finyr
IF inyr < startyr THEN
PRINT inyr; "  before start of data "
ELSE
        WHILE inyr > finyr
        READ name$, startyr, finyr
        ct = ct + 1
        WEND
        IF ct >= 12 THEN
                    PRINT "out of data"
        ELSE
        PRINT name$; " was PRESIDENT that year"
        END IF
END IF
END
DATA Rooseveldt,1933,1945
DATA Truman,1945,1953
DATA Eisenhower,1953,1961
DATA Kennedy,1961,1963
DATA Johnson,1963,1969
DATA Nixon,1969,1974
DATA Ford,1974,1977
DATA Carter,1977,1981
DATA Reagan,1981,1989
DATA Bush G.H.,1989,1993
DATA Clinton,1993,2001
DATA Bush G.W.,2001,2004
END
```

Examiner’s Guidance Notes

This was a popular question and usually done well. Most marks were lost for not expressing a data structure for the names and years from several alternatives, e.g. record/file, assignment, even ‘IF’ sequences. Some lacked any sort of search facility for the file or records. Some poor answers read in the name of the President and printed it out again.

Question 7

7. A useful series for the calculation of π is due to Gottfried Liebniz (1646-1716)

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13}\ldots)$$

a) Establish a relationship between successive terms of this series. **(3 marks)**

b) Use this in a program to compute π to 10 decimal place accuracy. Print how many iterations are needed: [π = 3.1415926535] **(9 marks)**

Answer Pointers

Development:

It can be seen that after the first two terms the denominator in the series is a regular sequence of odd numbers (3,5,7,11,…) which is calculable from (2n+1) where n = 1,2,3 and so on. The sign of the terms alternates between +1 and -1, easily determined from
sign [new] ← sign[previous] * (-1)
Hence the central loop will contain
oddno ← 2*n + 1
term ← 4 * (1/oddno) * sign
‘Ten decimal place accuracy’ implies that any number so small that it is less than
$0.5 * 10^{-11}$ will not change the value of the first 10 decimal places. Hence we can set accuracy to 0.5E-11. (3 marks)

Complete Pascal program

```
PROGRAM picalc(INPUT,OUTPUT);
CONST pi = 3.1415926535; accuracy = 5E-11;
VAR n, sign, oddno, cycles, disp:INTEGER; picalc, term, error,
prevterm:REAL;
BEGIN
WRITELN('LIEBNIZ method to calculate pi');
WRITELN('Maximum cycles? ');
READ(cycles);
n := 0; sign := -1; error:= 1.0; picalc := 4.0;
WHILE (error > accuracy) AND (n < cycles) DO
      BEGIN
      n := n + 1;
      oddno := 2*n +1;
      term := 4 * (1/oddno) * sign;
      picalc := picalc + term;
      sign := sign *(-1);
      error := ABS(picalc - pi);
      prevterm := term
      END;
WRITELN;
WRITELN('program ends : pi = ',picalc:12:10,'  ', n:4, ' iterations')
END.
```

(9 marks)

Examiner's Guidance Notes

Not a popular question. A few derived mathematical formulae for the result which were quite elegant for use in a straightforward program. Full marks were awarded for this. However it did not have to be done this way, as shown in the model answer above.

Many candidates worked out the value of the sequence of fractions, definitely NOT what was wanted; the question was not intended to be one on using a calculator! Other poor answers had '…' in the series, not realizing that you cannot imply operations like this in any computer program. There were many very weak answers here as such candidates did not grasp what the essentials of the answer required.

Question 8

8. While the area of an ellipse having major axis (a) and minor axis (b) is given exactly by the formula A = π*a*b there is no formula which gives the exact perimeter(S). The best approximation is given by Ramanjuan's formula:

$S \cong \pi(a + b)\ [1\ +\ 3t\ /\ (10 + \mathrm{sqrt}(4\ -3t))]$ where t is given by

$t = (a - b)^2 / (a + b)^2$

Other approximations are $S \cong \pi(\ a + b)$ and $S \cong 2*\pi*a*b$.

Write a program to compute the area for a group of ellipses having major axis (a) and minor axes between bmin and bmax, which are requested interactively. It also computes the perimeter by each of the three given formulae, and finds the percentage difference of the other two from that given by the Ramanjuan formula. **(12 marks)**

Answer Pointers

```
PROGRAM ellipse1(INPUT,OUTPUT);
{compares various formulae for the circumference of an ellipse, evaluates area}
CONST pi = 3.1415926535;
VAR a,b,bmax,bmin:INTEGER; Area,circ_1,circ_2,p1,p2,e,t,S:REAL;
BEGIN
WRITELN('ellipse circumference - comparison of formulae');
WRITELN('Ramanjuan     pi(a+b)    error    pi(2ab)');
WRITELN('input major axis [a]'); READ(a);
WRITELN('input largest minor axis (b-max)'); READ(bmax);
WRITELN('input smallest minor axis (b-min)'); READ(bmin);
WRITELN('a-axis b-axis  e  pi(a + b) percent  pi(!ab), percent  Ramanjuan
Area');
FOR b:= bmax DOWNTO bmin DO
      BEGIN
      e :=SQRT(a*a - b*b)/a;
      Area := pi*a*b;
      circ_1 := (a + b)* pi;
      circ_2 := 2*SQRT(a*b)*pi;
      t :=SQR(a - b)/SQR(a + b);
      S := pi*(a + b)*(1 + 3*t/(10 + SQRT(4-3*t)));
      p1 :=100*ABS(S - circ_1)/S;
      p2 :=100*ABS(S - circ_2)/S;
      WRITELN(a:3,' ':3,b:3,' ':3,e:6:3,' ':3,circ_1:6:3,' ':3,p1:4:1,
            ' ':3,circ_2:6:3,' ':3,p2:4:1,' ':6,S:6:3,' ':3,Area:6:3)
      END;
WRITELN('program ends')
END.
```

Examiner's Guidance Notes

This was obviously perceived as a difficult question as it attracted few answers. Marks were lost for failure to have prompt captions for essential input (a, bmax, bmin); no loop to range over the values between <bmax> and <bmin> for the different calculations; Inability to translate the given expressions for S and t with no implied operations e.g. 3t→3*t;
$(a – b)^2$ → sqr(a – b) or (a – b) * (a – b); incorrect percentage calculations.
A caption to the output table was also expected.
A more subtle error was using a real value (for b) to control a 'FOR' loop. (This is allowed in QBASIC). It was not penalized. Better to use 'WHILE' with a condition b < bmax to control iterations.

Question 9

9. When a user is entering data on an interactive form there are several interface elements that can be used for different kinds of data. Summarise the elements, show what they look like on the interface and describe the type(s) of data for which they are best used. **(12 marks)**

Answer Pointers

drop-down listbox/menu/select/combo (can be largely hidden apart from one choice until clicked on)
choose 1 from many or several from many, a lot of choices taking up little screen area

radio buttons (circles)
choose exactly 1 from many

check boxes (squares)
choose none, one or several from many

text box (rectangle, 1 line)
a little text (can be made secret for password)

textarea (rectangle, multi-line)
a lot of text

Examiner's Guidance Notes

(name , drawing , use)=3 marks per element; for any 4 elements; 12 marks total

It was not necessary to draw a complete web page. Just the drawings of the interface elements on their own was sufficient. However it was acceptable to place all the drawings together on a fictitious web page provided the drawings were labelled.

The question asked about interface elements that can be used for entering data on an interactive form. Therefore the marks were given for just those form elements that are active during data entry. Marks were not awarded for inactive elements (like labels) or for structures outside the form (like a frame).

An answer which gave "to make choices" as the use of menus or radio buttons or check boxes was not acceptable. It was necessary to be more specific - e.g. choose exactly 1.

Several candidates misunderstood the question and started describing various programming datatypes like integer, real, char, string, etc. A small number of candidates thought the question was about different types of interface and started describing the differences between GUI & command line.

Question 10

10. *a)* Suppose that marks are available for students as follows: Anne 40, Colin 55, Daniel 44, Frank 60, George 52, Harry 70, Jill 62, Kate 39, Lynne 74. This data is to be stored in a sequential file.

i) Draw a diagram representing the sequential file.

ii) List in order all the keys that will be visited to lookup the marks for the student Harry. **(4 marks)**

b) Now suppose that the file is made into an indexed sequential file with a single level index. The index groups are defined by initial letter of name, with 3 initial letters per group, i.e. A-C, D-F, G-I, etc.

i) Draw a diagram representing the file and its index.

ii) List in order all the keys that will be visited to lookup the marks for the student Kate indicating which keys are accessed in the index and which in the main file. **(8 marks)**

Answer Pointers

(a)(i)

1	Anne	40
2	Colin	55
3	Daniel	44
4	Frank	60
5	George	52
6	Harry	70
7	Jill	62
8	Kate	39
9	Lynne	74

(ii)
Looking for Harry, keys:
Anne, Colin, Daniel, Frank, George, Harry

(b)(i)

Group range for index	Last key in group	First - last record in group		Record number, sequential file	Key	Mark (%)
A-C	Colin	1 - 2		1	Anne	40
D-F	Frank	3 - 4		2	Colin	55
G-I	Harry	5 - 7		3	Daniel	44
J-L	Lynne	7 - 9		4	Frank	60
				5	George	52
				6	Harry	70
				7	Jill	62
				8	Kate	39
				9	Lynne	74

(ii) To find Kate...
(In Index:) Colin, Frank, Harry, Lynne
(In main file:) Jill, Kate

Examiner’s Guidance Notes

There were very few good answers to this question.

The question speaks about looking up the marks for a particular student. This means that the student name is being used as a key to find a mark. It was therefore very disappointing to find so many answers that turned this round and tried to work with the mark as the key. An all-too-common (incorrect) answer was one in which the records were first sorted into mark order.

Another typical problem with answers to this question was that mistakes were made with the alphabet. The question actually mentions the groups A-C, D-F, G-I yet there were answers with groups A-B, C-D, E-F, etc. (too few in group), answers with groups like A-C, D-F, H-J, K-M (missing letter), answers containing the group G-J (too many in group), answers which put Jill in group G-I, and so on.

The question clearly asks for a list of keys that will be visited to lookup Harry. This means that a list like (Ann 40), (Colin 55), ... is wrong because that is a list of records, not a list of keys. A list like 40, 55, 44, ... is wrong because the keys are the names, not marks. A list like Anne, Colin, Daniel, Frank, George is wrong (Harry not included) because you can't "find" Harry without visiting the key Harry.

Several answers tried to look up Harry in both parts (a) and (b) instead of Harry in part (a) and Kate in part (b).

Question 11

11. Consider the function min (see box below).

a) How would a black-box test of the function min be performed? **(3 marks)**

b) How would a white-box test of the function min be performed? **(3 marks)**

c) In what way would the error in the function show up under black-box testing using the test case min(4,2,3)? **(3 marks)**

d) In what way would the error show up under white-box testing using the test case min(4,2,3)? **(3 marks)**

```
int min(int a, b, c){
/* specification: find the minimum of 3 values given as parameters */
  int min; /* hold the minimum value found so far */
  min=a;
  if(b<a)
    min=b;
  if(c<a)
    min=c;
  return(min);
}
```

Answer Pointers

(a) Black-box testing takes place with an executable version of the module *min*. Test sets (perhaps with the minimum value in each of the three possible positions) are created and the module is run with test case input and the results inspected to see if they match with the expected results (obtained from the specification)

(b) White-box testing takes place with the source code of *min* available. Test sets are created and all paths through *min* are subjected to a dry run to see that the code performs as expected.

(c) Under black-box testing the result of running min(4,2,3) would be 3. The expected answer is 2 as defined by the specification of the function. Therefore the actual and expected results do not agree and an error has been found.

(d) [The comment about the purpose of the local variable min is critical in this part of the question]
Under white-box testing the function first considers a and assigns the value of 4 to min - so min can be said to contain the minimum of all the values considered so far - just 4
the first 'if' results in the assignment of 2 to min. So min can be said to contain the minimum of all the values considered so far (4,2)

Then the second 'if' results in the assigning of 3 to min. This new value of min is larger than the previous value and so it violates the meaning of the variable min as defined by the comment. So the error is discovered on executing the second 'if'.

Examiner’s Guidance Notes
The question asks how would testing of the function *min* be performed. An answer to this question which did not mention the function *min* could not be regarded as complete. Even the answers to parts (a) and (b) should be related to *min*.

In parts (b) and (d) many candidates wrote that white box testing would find syntax errors. But compilers do this - it is logic or semantic errors that need to be found.

In part (c) the candidate did not have an executable version of *min* to run to find the result from the input (4,2,3). However the code was simple and it was expected that the candidate would work out what the answer would be.

In part (c) as in many places elsewhere, care is needed to write down conclusions correctly. For example it is wrong to write in the answer that "the actual value returned is 3 which is false". The value 3 is an integer and so it cannot be the boolean false. A bit more care produces the answer "the actual answer (3) is compared with the expected answer (2) to see if they are equal - and the result of the equality test is false".

In part (d) the most popular comment was that the function compared *b* with *a* and *c* with *a* but did not compare *b* with *c* and that this was therefore the mistake. If students thought about finding the minimum of an array then a typical trace would be

```
min=v[0]
if(v[1]<min)min=v[1];
if(v[2]<min)min=v[2];
if(v[3]<min)min=v[3]; ... etc.
```

By comparison with this the correct code for function *min* would be

```
min=a;
if(b<min)min=b;
if(c<min)min=c;
```

Question 12

12. *a)* Consider the data shown below:

Account Number	11223344	22334455	33445566	...	...	...	...
Age	29	33	45	...	...	...	...
Name	J.White	A.Black	N.Brown	...	...	...	...

Imagine slicing this data up into horizontal rows and using three arrays to hold the data. How might the arrays be declared (in a programming language of your choice). How would the name and age details of a person with a particular account number be obtained? **(6 marks)**

b) Consider the data shown below:

Account no.	11223344	22334455	33445566	...	...	...	...
Age	29	33	45	...	...	...	...
Name	J.White	A.Black	N.Brown	...	...	...	...

Imagine slicing this data into vertical columns and using a database table to hold the data. Describe the design of the table. How would the name and age details of a person with a particular account number be obtained? **(6 marks)**

Answer Pointers

(a)

```
const int size=100; /* or some other number */
int   accountNo[size];
int   age[size];
char  name[size][7]; /* names are at least 7 chars long */
```

Note that the dimension of all the arrays is the same
[or equivalent in other languages]

to find data

```
for(i=0; i<size; i++){
      if(accountNo[i]==given)
            posn=i;
}
printf("%d", age[posn])
printf("%s", name[posn])
```

(b)

Simplest way to answer is to give SQL commands

Create table with specified fields...

```
CREATE table employee(
      accountNo VARCHAR[8],
      age INTEGER,
      name VARCHAR[25]
)
```

Obtain name and age given an account number...

```
SELECT name, age FROM employee WHERE accountNo='given'
```

Examiner’s Guidance Notes

(a) The question specifically asks for a solution with 3 arrays, therefore a solution with a single array of records with 3 fields per record did not obtain full marks

Many answers missed the 2-dimensionality of name giving char name[size]; or equivalent in other programming languages, thus restricting name to one character.

(b) a programming solution with a single array of records with 3 fields per record was acceptable here in place of the SQL answer.