FETAC
Further Education and Training Awards Council
Comhairle na nDámhachtaíní Breisoideachais agus Oiliúna

(National Council for Vocational Awards)

Computer Programming C20013

Theory Examination 2003

Duration: Two Hours

INSTRUCTIONS TO CANDIDATES:

*Answer any **three** questions*

All questions carry equal marks

Return this exam/answer paper when finished

Extra paper is available from the exam supervisor if required

This written exam counts as 40% of the total module

NAME (PRINT): Worked Solution

EXAM NUMBER

Question 1. Total 40 marks.

(a) This program contains 5 errors that will stop it from compiling. List the errors. ***20 marks***

```
#include <stdio.H>
main ()
{
 int looper;
 printf ("These are the first 10 squared numbers:\n);
 loopy = 1;
 while (looper <= 10)
 {
  printf ("%d\n", looper * looper);
  looper++;
 }
}
```

1	Should be lowercase
2	Declared and used variable names are different
3	Inverted commas not closed
4	Semicolon missing at end of line
5	Extra Right bracket present

(b) What is a variable used for? ***10 marks***

To store data; may be changed as the program runs

(c) What is the difference between a character and a string variable? ***10 marks***

Char stores one character. String holds many characters - a word, for example

Question 2. Total 40 marks.

(a) What type of numeric data should not be stored in the **int** data type? What data type should be used instead? ***10 marks***

Decimal values should not be stored in the int data type. Float should be used instead

(b) Write the general form of the **if** statement: ***10 marks***

```
if ( condition)
{ action 1 }          // condition = TRUE
else
{ action 2 }          // condition = FALSE
```

(c) Write a C program containing a loop that writes out the odd numbers between 9 and 99 ***20 marks***

must be complete

```
#include <stdio.h>
main()
{
int loop_var;
  loop_var = 1;                    // start at 1
  while (loop_var <= 99)
  {
    printf ("%d\n", loop_var;
    loop_var = loop_var + 2;       // go up by two
  }
}
```

Question 3. Total 40 marks.

(a) Draw a diagram to represent the state of the **numbers** array after this program finishes. ***30 marks***

```
#include <stdio.h>
main ()
{
 int numbers[9], loopvar;
 loopvar = 0;
 while (loopvar <= 9)
 {
  numbers[loopvar] = 100 - (loopvar * loopvar);
  if (loopvar == 5)
  {
   numbers[loopvar] = 0;
  }
  loopvar++;
 }
}
```

0..9 = 10 elements

loopvar goes from 0 to 9

each element of array is 100 - square of loop var

an exception

Draw your diagram here:

Numbers Array:

0	1	2	3	4	5	6	7	8	9
100	99	96	91	84	0	64	51	36	19

100 100 100 100 100 100 100 100 100
-0^2 -1^2 -2^2 -3^2 -16 -36 -49 -64 81

There is only one number written out.

(b) What numeric screen output is generated by this program line: ***10 marks***

```
printf ("%d\n", 'H'+'w');
```

use ASCii chart

'H' = 72
'w' = 119

72 + 119 = 191 ← answer

Question 4. Total 40 marks.

not a full program – write the relevant bits only.

(a) Write a C loop to read in an array of 20 numeric variables; then write another loop to write out the contents of the array in reverse order. Also write out the total sum of all the values in the array. ***30 marks***

```
total = 0;
loop = 0;
while ( loop <= 19)
{
   scanf ("%d", &MyArray[loop]);
   total = total + MyArray[loop];
   loop ++;
}
loop = 19;
while ( loop >= 0)
{
   printf ("%d", MyArray[loop]);
   loop ++;
}
printf ("Total is: %d\n", total);
```

(b) The control variable for a **while** loop should appear in a program not less than four times. List those times. ***10 marks***

1	Declare
2	Initialize → give a starting value
3	Compare → is the loop over?
4	Progress → get nearer to the end

Figure 1. The ASCII table.

				032	SP	033	!	034	"	035	#
036	$	37.00%		038	&	039	'	040	(	041	)
042	*	043	+	044	,	045	-	046	.	047	/
048	0	049	1	050	2	051	3	052	4	053	5
054	6	055	7	056	8	057	9	058	:	059	;
060	<	061	=	062	>	063	?	064	@	065	A
066	B	067	C	068	D	069	E	070	F	071	G
072	H	073	I	074	J	075	K	076	L	077	M
078	N	079	O	080	P	081	Q	082	R	083	S
084	T	085	U	086	V	087	W	088	X	089	Y
090	Z	091	[	092	\	093	]	094	^	095	_
096	`	097	a	098	b	099	c	100	d	101	e
102	f	103	g	104	h	105	i	106	j	107	k
108	l	109	m	110	n	111	o	112	p	113	q
114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}
126	~	127	□								

Printable alphanumeric and punctuation characters used in normal document text

Rough Work Page