



**(National Council for Vocational Awards)**



# **Computer Programming C20013**

**Theory Examination 2004**

## **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

PPS NUMBER: /

DATE: /

**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 ()
{
    char name[20];
    int age ; // Error 1: Missing semicolon
    printf ("Please enter your name and age:");
    scanf ("%s %d", &name, &age); // Error 2: Missing comma between variables
    printf ("Hello, %s\n");
    if (age > 30)
        printf ("You are ancient!\n"); // Error 3: Missing closing brace
    else
        printf ("Pass the Pampers!\n");
}
```

1	<stdio.h> should be on first line
2	Semi-colon on line int age;
3	Comma missing between variables in scanf statement
4	It should be a scanf statement!
5	Low inverted commas not closed.

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

To store data during the running of  
the program. The data may be changed as needed.

(c) What is the difference between an integer and a float variable? Give a sample of each type of data. **10 marks**

Integer is for whole numbers only e.g. 147

Float is for decimal numbers e.g. 3.14

**Question 2. Total 40 marks.**

- (a) What form of character data should *not* be stored in the **char** data type? How should such data be stored instead? **10 marks**

Character data with multiple letters. This should be saved as a string instead.

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

```
if ( condition )
{ action 1 }
else { action2 }
```

- (c) Write a C program containing a loop that writes out the even numbers between 10 and 100

**20 marks**

```
#include <stdio.h>
main ()
{
    int myloop;
    myloop = 10;
    while ( myloop <= 100 )
    {
        printf ("%d \n", myloop);
        myloop = myloop + 2;
    }
}
```

**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[6], loopvar;
    loopvar = 0;
    while (loopvar <= 5)
    {
        numbers[loopvar] = 500 - (loopvar * loopvar * loopvar);
        if (loopvar == 3)
        {
            numbers[loopvar] = 500 + loopvar * 2;
        }
        loopvar++;
    }
}
```

eg  $500 - 0^3 = 500$   
 $500 - 1^3 = 499$   
 $500 - 2^3 = 492$   
 $500 - 3^3 = 492$  (64)  
 $500 - 4^3 = 375$  (125)

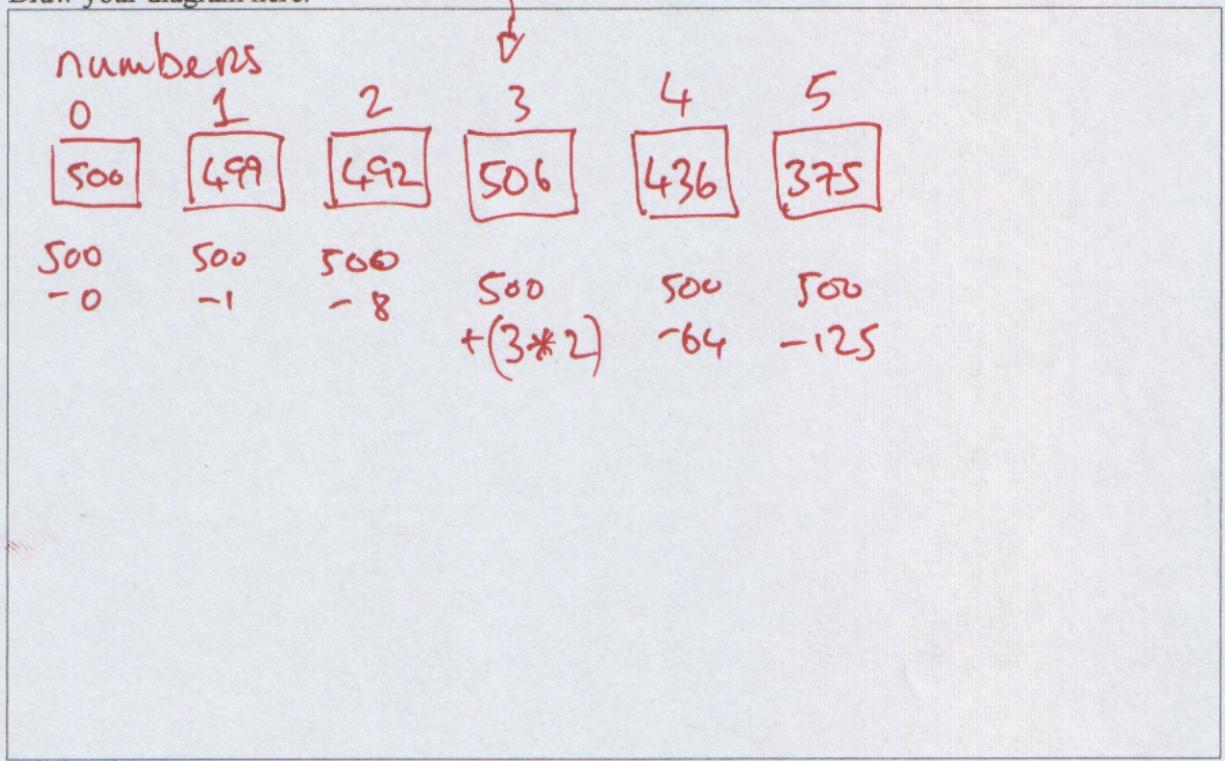
*0 → 5 elements*

*500 - loopvar cubed*

*This breaks the rule*

*3*

Draw your diagram here:



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

```
printf ("%c%c%c%c%c%c%c%c\n", 71,111,111,100,32,76,117,99,107);
```

→ Must use the ASCII char

↓  
page 6

Good Luck

Space

**Question 4. Total 40 marks.**

not a full program. I can leave some bits out!

- (a) Write a C loop to read in an array of 30 numeric variables; then write another loop to write out every second element of the array. Also calculate the average of *all* the values in the array.

Starting bits  
left out

30 marks

```

int loop;
loop = 0; total = 0;           // declare
while (loop < 30)             // declare initialize
{                           // compare
    scanf ("%d", &numlist[loop]);
    loop++;                  // ← progress
    total = total + numlist[loop];
}
loop = 0;
while (loop < 30)
{
    printf ("%d\n", numlist[loop]);
    loop = loop + 2;          // up by 2 to do
}                           // every second one
printf ("Average is %f\n", total / 30);

```

program ends

- (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
3	compare
4	progress (change the variable so the loop ends eventually)

**Figure 1. The ASCII table.**

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

Printable alphanumeric and punctuation characters used in normal document text

***Rough Work Page***

---

***Rough Work Page***

---