



University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER I RE-SIT EXAMINATION – 2010/2011

COMP 10060

Computer Science for Engineers

Prof. M. McGinnity

Prof. J. Carthy

Prof. J. Murphy*

Time Allowed: 2 Hours

Instructions for Candidates

Answer **Question 1** and **one** other Question.

Question 1 carries 60 marks; Questions 2 and 3 carry 40 marks.

READ EACH QUESTION CAREFULLY.

Instructions for invigilators

This is a closed-book examination. No calculators allowed.

Loose Rough Work sheets are not to be distributed or used.

Question 1 (COMPULSORY) [60 marks]

Answer all parts (a) – (t). Each part carries 3 marks.

(a) True or False (*no explanation required*): In C, all declarations and statements **must** end with a “:”

(b) True or False (*no explanation required*): In C, the programmer can decide where the program will start execution, the default being **main()**

(c) True or False (*no explanation required*): “**three float**” is a **valid** identifier in C.

(d) True or False (*no explanation required*): If a program compiles and produces an executable, then this **can** still have errors when executed.

(e) Complete the sentence: “The **two** most widely used tools for developing algorithms are _____ and _____”

(f) If **x==9**, **y==6**, and **z==6**, what is the value of **w** in the expression **w = x / y * z** ?

(g) What is the screen output of the following fragment of C code (*no explanation required*)?

```
int a=1, b=3, c=-1;
if ((a-c) && ((b/3)>0) ){
    printf("condition is true\n");
} else {
    printf("condition is false\n");
}
```

(h) What is the screen output of the following fragment of C code (*no explanation required*)?

```
int i=30;
while (i>0){
    i = (i-1)/3;
    printf("i is %d\n",i);
}
```

(i) What is the screen output of the following fragment of C code (no explanation required):

```
float x=-0.00293607;
printf("value is %.2f\n",x);
```

[Question 1 continues]

Question 1 (continued)

(j) What is the screen output of the following fragment of C code (*no explanation required*)?

```
int i=-5,j=5;
if (j<=-10){
    printf("first\n");
} else if ((-i)>=(-j)){
    printf("second\n");
} else {
    printf("no match\n");
}
```

(k) What is the screen output of the following fragment of C code (*no explanation required*):

```
int i;
for (i=1;i<6;i++){
    switch(i)
    {
        case 2:    printf("2\n");
        break;
        case 3:    printf("3\n");
        default:   printf("default\n");
    }
}
```

(l) What is the screen output of the following C program (*no explanation required*)?

```
#include <stdio.h>
int f1(int a, int b, int c){
    return (a+b*c);
}
int f2(int a, int b){
    int c = f1(a, b, 3);
    return c+1;
}
int main(void)
{
    printf("result is %d\n", f1(-1,f2(2,4),2));
    return 0;
}
```

(m) What is the screen output of the following fragment of C code (*no explanation required*)?

```
int x = 3;
int y = -2;
int* p = &x;
*p = (*p)*x + (*p)*y;
printf("x is %d and y is %d\n", x, y);
```

[Question 1 continues]

Question 1 (continued)

(n) Select the correct answer: If the pointer **ptr1** currently points to **x** and you want to assign the current value of **y** to **x**, you could use the statement

- (1) **ptr1 = &y;**
- (2) ***ptr1 = y;**
- (3) **y = *ptr1;**

(o) Select the **correct** answer: If you want to multiply the second and fifth elements of an array **arr[]** together, you should use

- (1) **arr[2]*arr[5]**
- (2) **(* (arr+1)) * (* (arr+4))**
- (3) ***arr[2]**arr[5]**

(p) What is the screen output of the following fragment of C code (*no explanation required*)?

```
char str[]="abcdefghijklmn";
char nonvowels[]="bcd fghjklmn";
int i, j;
for (j=0; nonvowels[j]!='\0'; j++){
    for (i=0; str[i]!='\0'; i++){
        if (str[i]==nonvowels[j]){
            str[i]='N';
            break;
        }
    }
}
printf("string=%s\n", str);
```

(q) True or False (*no explanation required*): If **i** and **j** are integer variables and **arrflts[]** is an array of **floats**, **arrflts[i+j++]** is a valid expression in C.

(r) Select the correct answer: **fopen("something.txt", "r")** means

- (1) if it exists, open **something.txt** for random access, otherwise return an error.
- (2) if it exists, open **something.txt** for reading only, otherwise return an error.
- (3) if it exists, open **something.txt** for reading only, otherwise open the first file found and read from it.

(s) The function prototype

```
void fn(int x);
```

tells us that **fn** takes what argument and type (if any) and returns what value and type (if any).

(t) True or False (*no explanation required*): in C, the **name** of a string is treated by the compiler as a pointer-to-char variable whose value can be reassigned in the program.

Question 2 [40 marks]

Answer all parts (a) – (c).

- (a) The following fragment of C code outputs the lowercase letters in a string called **string** to the screen using a **while** loop:

```
int i=0;
char string[100];
/* assume string[] is somehow filled with letters, digits, etc
*/
while (string[i]!='\0'){
    if ((string[i] >= 'a') && (string[i] <= 'z')){
        printf("%c", string[i]);
    }
    i++;
}
```

Re-write this code fragment using a **for** loop instead of the **while** loop.

[14 Marks]

- (b) The following fragment of C code uses **if** statements to determine whether to Sell, Hold, or Buy a stock based on its price:

```
if (price>=20){
    printf("Sell\n");
}
if ((price>=10) && (price<20)){
    printf("Hold\n");
}
if (price<10){
    printf("Buy\n");
}
```

Re-write this code fragment using **if/else-if/else** statements instead of the **if** statements.

[14 Marks]

- (c) Consider the following C program:

```
#include <stdio.h>
/* DEFINITION OF FUNCTION "numneg" GOES HERE */
void main(void) {
    int i, j=0, array1[8]={1,-1,-1,1,0,1,0,-1};
    j = numneg(array1,8); /* function call to numneg() */
    printf("there are %d negative elements of array1[]", j);
}
```

Write down the definition of function **numneg()** which counts the number of negative elements in its input array, so that the output of the above program is:

there are 3 negative elements of array1[]

[12 Marks]

Question 3 [40 marks]

Answer parts (a) and (b).

(a) Consider the following C program:

```
#include <stdio.h>
void main(void)
{
    int intarr[3], i;
    for (i=0 ; i<3; i++){
        printf("enter value number %d: ", i+1);
        scanf("%d", &intarr[i]);          /* LINE 1 */
        printf("you entered %d\n", intarr[i]); /* LINE 2 */
    }
}
```

Re-write the lines **LINE 1** and **LINE 2** using “array pointers” instead of array subscripts.
[20 Marks]

(b) Consider the following C program:

```
#include "stdio.h"
void main(void){
    char message[80]="hello there";
    int count=0;
    while (message[count]!='\0'){
        count++; /* increase by 1 as long as NULL not reached */
    }
    printf("\"%s\" has length %d\n", message, count);
}
```

(i) What is the screen output of this program (*no explanation required*)?

(ii) *Re-write* the above program so that all the code for determining the length of the string **message** is contained in a function called **stringlength()** that you should define.

[20 Marks]