GLASGOW COLLEGE UESTC

Main Exam paper

Course name (UESTC 1005 – Introductory Programming)

Date:30th, December
Time: 19:00-21:00

Attempt all PARTS. Total 100 marks

Use one answer sheet for each of the questions in this exam.
Show all work on the answer sheet.

Make sure that your University of Glasgow and UESTC Student Identification Numbers are on all answer sheets.

An electronic calculator may be used provided that it does not allow text storage or display, or graphical display.

All graphs should be clearly labelled and sufficiently large so that all elements are easy to read.

The numbers in square brackets in the right-hand margin indicate the marks allotted to the part of the question against which the mark is shown. These marks are for guidance only.

Continued overleaf

Q1 (a) Hao works for a construction company and has been asked to design a football stadium. For that, she has written C programs to perform preliminary survey. However, the results are unexpected and wrong. Can you identify and fix the errors below?

(i) The aim is to calculate the area of a triangle having sides `a` and `b`:

```
1    double a, b;
2    printf("Enter the length a:");
3    scanf("%d", &a);
4    printf("Enter the length b:");
5    scanf("%f", &b);
6    printf("The area is %lf",(1/2)*a*b);
```

Figure Q1(a)(i)

[5]

(ii) In the stadium design, Hao needs to determine the distance between the centre of the football pitch and the seats in stadium. To do so, write a C program that that accepts from the user two real-valued points in the three dimensional space and calculates the distance. The distance formula is:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$ [5]

(b) Write down the output of the program in Figure Q1(b). [5]

```
#include <stdio.h>

int main()
{
    int count = 0;
    signed char c = 127;
    count = ++c;
    printf("count=%d\n", count);
    return 0;
}
```

Figure Q1(b)

(c) Write a C program to read 5 integers from keyboard and find their sum and average.
An example (with user inputs in bold text) is shown in Figure Q1 (c).

```
Example:
Input the 5 integers:
Number-1 :2
```

Continued overleaf

```
...
Number-5 :2

The sum is : 10
The average is : 2.000000
```

Figure Q1(c)

(i) The program should accept an input, and display the output in the way shown in Figure Q1(c) [5]

(ii) Demonstrate the correct use of loop in the program [5]

Q2 (a) In the UK, the frequently used banknotes are valued at £5, £10, £20, and £50, and the commonly used coins are valued at 1p, 5p, 10p, 50p and £1. Note that 100p = £1. Yang goes to a bank in the UK and wishes to withdraw money in his account in the form of banknotes only. However, he wants the smallest number of bank notes and does not want any coins.

Write a program that accepts as an input from the user any amount in pound sterling (£) and breaks down the money in the smallest number of banknotes. The program should also display the remaining amount of money left after the withdrawal.

As an example, if the user inputs the amount 55.65, the result should be, 1 £50 banknote, 1 £5 banknote, and the remaining amount £0.65. Another example, if the user enters 174.25, the result should be, 3 £50 banknotes, 1 £20 banknote and 4.25 as the remaining amount. [10]

(b) Write a program in C to find the prime numbers within a range of numbers. An example (with user inputs in bold text) is shown in Figure Q2 (b).

```
Example:
Input starting number of range: 1
Input ending number of range : 30

The prime number between 1 and 30 are :
2 3 5 7 11 13 17 19 23 29
```

Figure Q2(b)

(i) Write the part of program accepting the input and generating the output. [3]

(ii) Write pseudocode showing the program structure to obtain prime numbers within a given range. [5]

Continued overleaf

(iii) Write the part of program based on the pseudocode to form a complete program realising the required function. [7]

Q3 Write a C program to add two 1D arrays which are of the same length (each of them is no longer than 100) and sort the output in descending order. The array sorting must be put into a function and is called by the main function, which carries out all the other operations.
You will ask the user the length of the arrays and input them.

```
Example:
        Input the number of elements in either array: 3
        Input the 3 elements in the first array:
        element[0]: 1
        element[1]: 2
        element[2]: 3
        Input the 3 elements in the second array:
        element[0]: 4
        element[1]: 5
        element[6]: 6

        The sum in descending order is 9 7 5
```

Figure Q3

(a) Write the code of the main program that inputs the two arrays and displays the result. [5]

(b) Write the code of the main program that manipulates the two arrays and sum them. [5]

(c) Complete the function of array sorting using pointers

(i) Write the function definition and prototype. [5]

(ii) Write the pseudocode of the array sorting function showing the program structure. [5]

(iii) Write the program based on the pseudocode resulting in a complete array sorting function with pointers. [5]

Q4 (a) Examine the program shown in Figure Q4(a) where the code snippet should swap the text stored in string variables `str1` and `str2` and subsequently print it. However, there is an error in the code and the program fails to swap the values of `str1` and `str2`.

```
1       #include<stdio.h>
2       void swap (char * strl, char* str2){
3             char * temp = strl;
```

Continued overleaf

```
4               str1 = str2;
5               str2 = temp;
6         }
7
8         int main ( ){
9               char * str1 = "Hello";
10              char * str2 = "UESTC";
11              swap (str1, str2) ;
12              printf ("str1 is %S, str2 is %s",str1, str2) ;
13              getchar();
14              return 0;
15  }
```

Figure Q4(a)

(i) Explain in your own words the error in the program [5]

(ii) Rewrite the `swap` function using pointers to swap the values of string variables `str1` and `str2`. [7]

(iii) Explain in words why we cannot use `strcpy()` function for string variables `str1` and `str2`. [3]

(b) Given the program in Figure Q4(b)

```
1         #include <stdio.h>
2         #define SIZE 10
3
4         int mystery ( const int b[], int p) ; /* function
          prototype */
5
6         /* function main begins program execution */
7         int main ()
8         {
9           int x; /* holds return value of function mystery */
10
11          /* initialize array a */
12          int a[ SIZE ]= {1, 2, 3, 4, 5, 6 , 7, 8, 9, 10 };
13
14          x = mystery ( a, SIZE ) ;
15          printf ( "Result is %d n", x ) ;
16          return 0; /* indicates successful termination */
17        }/* end main */
18        /* what does this function do? */
19        int mystery ( const int b[],int p)
20        {
21
22          /* base case */
23          if(p == 1 ) {
```

Continued overleaf

```
24          return b[ 0 ];
25        }/* end if */
26
27        else {
28          return b[ p - 1 ] + mystery ( b, p - 1 );
29        }/* end else */
30
31      } /* end function mystery */
```

Figure Q4(b)

(i) Explain in your own words what is recursion and state the line in the code where recursion takes place. [3]

(ii) Write down the output of the program [4]

(iii) Write down the output of the program if at line 2 we change the value of SIZE from 10 to 5. [3]

End of question paper