

GLASGOW COLLEGE UESTC

Exam paper

Course name (UESTC 1005)

Date: 3rd Jan. 2020

Time: 09:30-11:30am

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.

Q1 (a) For each sub-question below, write a one-line statement in the C programming language:

(i) Write the prototype statement for a function named `Calc` that takes as an input argument; one integer variable, one double variable, and a float type variable. The function `Calc` returns a char value. [3]

(ii) Use the placeholder in the `printf` function so that only two digits are displayed after the decimal point for the floating-point value 3.1416. [2]

(iii) Write an if condition statement to indicate whether a variable `x` is both greater than or equal to 100, and less than 500. [2]

(iv) Declare a pointer with an identifier `myPtr` and assign it to the address of the integer variable `x`. [2]

(b) Identify the errors in the following pieces of code and provide a reason for each error. Keep in mind that there may be syntax errors, logical errors, or both.

(i) The piece of code below is intended to read and print three integer variables:

```
int x, y, z;
scanf("%d %d %d", &x, &y, z);
printf("%2.2d %3f %f", x, y, z);
```

 [4]

(ii) The program below is intended to display elements of an integer array:

```
#define<stdio.h>;

float main();
{
    int _array[3], i;
    _array[3] = {'2', 1, -1};
    for (i = -1; i <= 3; i++)
    {
        printf("%d\n", _array[i]);
    }
    return 0;
}
```

 [5]

(iii) The statements below define integer variables and assign values:

```
int x, y, z;
x = 5; y = 3; z = 1;
y = ++x + y; z = z--> x;
```

 [2]

Continued overleaf

(iv) The program below computes the square of a number:

```
#define<stdio.h>
```

```
void main()
```

```
{
```

```
    int x, y;
```

```
    scanf("%d", &x);
```

```
    y = square(x);
```

```
    printf("%d", y);
```

```
    return 0;
```

```
}
```

```
float square(int x){
```

```
    return x*x;
```

```
}
```

[5]

Q2 (a) Write the output of the program in Figure Q2-a. [10]

(b) Write down the sequence of values that the counter variable `i` obtains when the outermost for loop is executed in Figure Q2-a. [5]

(c) Explain briefly what are linked lists? [5]

(d) Write two advantages and two disadvantages of linked lists? [5]

Q3 (a) Write a program in which the user enters two dates from the keyboard and then compares them. To store each date use **structure** that contains three members namely date, month, and year. Display the message "Equal" if the dates are equal, otherwise "Unequal". [10]

(b) Write a program that produces the following output:

The sum of x and y is 13

The program should define a macro `SUM` with two arguments, `x` and `y`, and use `SUM` to produce the output. [10]

(c) Consider a two-dimensional integer array `t` of size 2×5 . Write a nested for statement that initializes each element of `t` to zero. [5]

Q4 (a) Write a function named `CoinToss` to simulate a coin toss: Create a variable that gets '0' or '1' with 50% probability each time you call the function. When the value of the variable is '1', you should display, "It's Heads", and when '0', it displays "It's Tails!". The function should also be able to return the value of the variable back to the main program. *Hint: you may use the random number generating function `rand()`.* [10]

Continued overleaf

```

#include <stdio.h>

int main() {
    int i, j, k, l, m;
    k = 3;
    l = 1;
    m = k - 1;
    for(i=1; i<k*2; i++)
    {
        for(j=1; j<=m; j++) {
            printf(" ");
        }
        for(j=1; j<l*2; j++) {
            printf("*");
        }
        printf("\n");
        if(i<k) {
            m--;
            l++;
        }
        else{
            m++;
            l--;
        }
    }
    return 0;
}

```

Figure Q2-a

- (b) Write a program that displays a triangle as shown in Figure Q4-b. Request the user to input the number of * characters that will be displayed at the top side, which in this case is equal to 7. Use nested loops to print the triangle sequence.

```

*****
*****
*****
****
***
**
*

```

Figure Q4-b

[6]

- (c) Write a function `Kg2lb` that takes weight in kilograms as an input, converts it from kilograms to pounds, and returns the value back to the main program. Note

Continued overleaf

that $1\text{ kg} = 2.2046\text{ pounds}$. Use an appropriate data type for input arguments and the return value. [5]

- (d) Write down the function `kg2lb` call statement that will convert 100 pounds and store the output in a variable, `weight_kg`. [2]
- (e) Write a statement that prints the value of the variable, `weight_kg` and displays only two digits after the decimal point. [2]