ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION
DURATION: 3 HOURS

WINTER SEMESTER, 2022-2023
FULL MARKS: 150

CSE 4107: Structured Programming I

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

---

1. a) Suppose you have a file named "numbers.txt" containing exactly 100 positive integers in separate lines. You wish to separate out the even and odd numbers into two separate files named "even.txt" and "odd.txt", respectively. 15 (CO3) (PO1)

   Write a C program that accomplishes this objective.

   b) What output is produced by the program given in Code Snippet 1? Show the entire process of generating the output by tracing each call to `recurse()`. 10 (CO3) (PO2)

```c
#include<stdio.h>

int recurse(int n){
    if(n==1) return 1;
    else if(n%2) return n + recurse(n/2);
    else return n + recurse(n-1);
}

int main(){
    printf("%d", recurse(12));
    return 0;
}
```

Code Snippet 1: A C program for Question 1.b)

2. a) Describe the two methods by which an array of strings may be defined. Provide justifications with proper examples showcasing which one of these two methods is more efficient. 4 + 6 (CO2) (PO1)

   b) Suppose you are trying to buy candies where each candy has a fixed price. Each candy gives you one empty wrapper. The store has a policy that you can return a fixed number of wrappers. You can keep doing this again as long as you have enough wrappers to exchange for more candies. You are given an integer representing *money* that you can use to buy the initial number of candies. The price of each candy is also an integer called *price*. The number of wrappers you need to exchange for one candy is another integer called *wrapper*. 15 (CO3) (PO1)

   Write a recursive function that takes *money*, *price* and *wrapper* as parameters and returns the maximum number of candies you can buy.

3. a) Write a C program that takes an integer n as input and prints 1 if it is divisible by 15 and prints 0 otherwise. The integer may be as large as 100 digits. 13 (CO3) (PO1)

b) Find the output of the program in Code Snippet 2. Also, mention which element of the `digits` array is pointed to (at line 13) by the pointer `p` at each iteration of the loop.

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int temp, digits[10] = {0, 1, 2, 3, 4, 5 , 6, 7, 8, 9};
6      int *p = digits;
7
8      while(1){
9          temp = *p;
10         *p = *(p+1);
11         *(p+1) = temp;
12
13         if(*p == 7)
14             break;
15         p++;
16     }
17
18     for(int i=0;i<10;i++){
19         printf("%d ", digits[i]);
20     }
21     return 0;
22 }
```

Code Snippet 2: A C program for Question 3.b)

4. Suppose you have a file named "employees.txt" that contains the information of 120 employees of a particular organization. The information for each employee is stored in separate lines in the file. Each line contains four information: `id`, `name`, `salary`, and `department` each separated by a whitespaces. The `id` is an integer. The `name` can have a maximum length of 30 alphanumeric characters. The `salary` is stored in dollar currency as a floating point value. The `department` can be one of the following: "Admin", "HR", "Accounting", and "Sales", which your organization has decided to enumerate as 1, 2, 3, and 4, respectively.

   a) Create an Enum as `enum Dept` and a structure as `struct Employee` for storing all the relevant information. Note that the structure must contain the `enum` object as a member. — 8 (CO3) (PO1)

   b) Write a C program that reads the "employees.txt" file and stores the information in an array of structure objects. Then the program prints the name of the employee having the highest salary among those working in the "HR" department. — 7+10 (CO3) (PO1)

5. a) Provide a comparative analysis among Structures, Unions, and Enums in C. — 6 (CO2) (PO1)

   b) Write short notes on the following topics: — 4 × 2 (CO1) (PO1)
      i. Command line arguments
      ii. `strcmp()` function
      iii. `fseek()` function
      iv. Null character

Find the output of the program given in Code Snippet 3. 11 (CO3) (PO2)

```
#include<stdio.h>
#include<string.h>

void print(char **y){
    int i;
    char *x=*y;
    *(x+4) = x[3];
    x += strlen(x);
    for(i=0; i<=5; i++){
        printf("%s\n",x);
        --x;
    }
}

int main(){
    int i;
    char *y, a[]="Apple";
    y = a;
    print(&y);
    return 0;
}
```

Code Snippet 3: A C program for Question 5.c)

6. a) Write a function that takes an array and the size of the array as parameters. It returns a pointer to the largest element in the array. The function prototype is given as follows: 8 (CO3) (PO1)

```
int *find_largest(int a[], int n);
```

b) The C program in Code Snippet 4 contains some errors. Identify the errors by stating the corresponding line number and mention the corrected version of the erroneous line. 10 (CO2) (PO1)

```
#include<stdio.h>

int main(){
    int a[];
    int b[1+2];
    int c[3, 3];

    for(i = 0; i=<2; i++){
        b[i] = i;
        a[i] = 1/i;
        c[i][i] = i*i;
    }
    return 0;
}
```

Code Snippet 4: A C program for Question 6.b)

c) Explain the differences between `gets()` and `scanf()` functions in C using an appropriate string(s) as sample input. 7 (CO1) (PO1)