United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam:: Trimester: Summer 2023

Course Code: CSE 1111, Course Title: Structured Programming Language

Total Marks: **40** Duration: **2 hours**

[Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.]

There are FIVE questions. Answer all the questions. Marks are indicated in the right margin.

Q.1 a) Write a **C program** according to the following: **[4]**

i) Write a function "***IsEven***" that will take an integer as parameter and determine **whether it is even or not**. If the integer is even, then the function will return 1. Otherwise, it will return 0.

ii) Write a function "***ComputeEvenSum***" that will take an array of integers and ***n*** as parameters. ***n*** is the number of values in the array. It computes the **sum of the even numbers** in that array and **returns the sum**. You **must use "*IsEven*" function** for even checking.

iii) In ***main( )*** function, **declare and initialize** the array and any other variables as needed. Then call the function "***ComputeEvenSum***" with appropriate parameters and finally **display** the **returned value**.

b) Find the **output** of the following program (left). Notice the **local and global contexts**. **[4]**

```
#include<stdio.h>
int a,b,c=5,d=2;
int func1(int a,int b){
   c=a+b;
   return c*2;
}
int func2(int p){
   p=p+a;
   a*=2;
   return p;
}
void func3(int d){
   int c=d+10;
   b=func2(c);
}
void main(){
   a=1;b=3;
   printf("%d %d %d %d\n",a,b,c,d);
   a=func1(a,b);
   printf("%d %d %d %d\n",a,b,c,d);
   d=func2(c);
   printf("%d %d %d %d\n",a,b,c,d);
   func3(b);
   printf("%d %d %d %d\n",a,b,c,d);
}
```

C Code for **1(b)**

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

void mystery(char *str, int d) {
   for(int i = 0; i < strlen(str); i++) {
      char c = str[i];

      if (c >= 'a' && c <= 'z') {
         str[i] = 'a' + (c - 'a' + d) % 26;
      }else if (c >= 'A' && c <= 'Z') {
         str[i] = 'A' + (c - 'A' + d) % 26;
      }
   }
}
void main() {
   char msg[50];
   strcpy(msg, "Eb");
   strcat(msg, "iil");
   mystery(msg, 3);
   puts(msg);

   strcpy(msg, "# Ayh T");
   mystery(msg, 6);
   puts(msg);
}
```

C Code for **2(a)**

Q.2 a) Find the **output** of the program **above to the right**. **[4]**

b) Write a **C program** that takes **two strings** from keyboard. The **first string** is a sentence and the **second string** is a single word. The program will find (case insensitively) the **number of occurrences** of the second string in the first string. **[4]**

Sample Input	Sample Output
First string: New experience is new learning **Second string:** new	2
First string: New experience is new learning **Second string:** nEw	2
First string: This is another example **Second string:** line	0

Q.3 Write a **C Program** to store the following information about international cricketers and find the "**Cricketer of the Year**": **[8]**

i) **Create a structure** named ***Cricketer*** with **name** (string of length 50), **country** (string of length 50), cricketer **type** (string of length 10), **wickets taken** in the **last 30 matches** (in an array), **runs scored** in the **last 30 matches** (in an array), **total match played** in the last year (int), and **overall performance score** (float).

a. There are **only 2 types** of cricketers: "***bowler***" and "***batsman***". So, the cricketer type in the structure stores only one of these two values.

ii) In the **main( )** function,

a. **Take input** for **100 Cricketers** from the user. DO NOT take the overall performance as input.

b. Additionally, **calculate the overall performance** of each cricketer in the following way:

i. If the cricketer is a "***bowler***": total wickets taken in the last 30 matches ÷ total matches played in the last year

ii. If the cricketer is a "***batsman***": total runs scored in the last 30 matches ÷ total matches played in the last year

c. **Find** and **display** the information of the "**Cricketer of the Year**" who has the **highest overall performance**.

Q.4 a) Find the **output** of the following program (left). **[4]**

```
#include<stdio.h>
void change (int *x, int *y, int z) {
   *x=*x+10;
   *y=*y+3;
   z=z+5;
 return;
}
void main(){
   int a=10, b=21, c= 40;
   printf("%d %d %d\n", a, b, c);
   change(&a, &b, c);
   printf("%d %d %d\n", a, b, c);
}
```

C Code for **4(a)**

```
void function(int num)
{
   if(num > 0)
   {
      function(--num);
      printf("%d ", num);
      function(--num);
   }
}
```

C Code for **4(b)**

b) If the function (given **above to the right**) is called passing an **integer value 4** as the argument, what will be the **resulting output**? **[4]**

Q.5 a) Write a **C program** that reads the numbers from the "***Sample.txt***" file given below, stores those values in an array, **calculates** the **sum of just those values that are even and divisible by 4**, and then **outputs the result** to the "***Output.txt***" file. A **Sample.txt** and **Output.txt** files are given below as an example. **[4]**

Sample.txt	Output.txt
16 -3 -4 7 12 2 .....more values	Sum: 24

b) **Complete** the following program to **compute the sum of all odd elements** in the array ***arr***. (Note: You **cannot use array index** like a[i]). **[4]**

```
#include <stdio.h>
int sumOfOddElements(int *arr, int size) {
   // Write your code here
}
int main() {
  int numbers[]={10, 21, 35, 42, 57, 68, 73};
  int n=sizeof(numbers)/sizeof(numbers[0]);
  int sum=sumOfOddElements(numbers, n);
  printf("Sum of odd elements: %d\n", sum);

   return 0;
}
```