National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus

0476

Course Name:	Programming Fundamentals	Course Code:	CS1002
Degree Program:	BS(CS) | BS(SE) | BS(AI)	Semester:	Fall 2023
Exam Duration:	140 Min	Total Marks:	40
Paper Date:	Monday, December 18, 2023	Obtained Marks	
Sections:	ALL	No of Page(s):	9
Exam Term & Type:	Final Term I Closed Book	Required Answer Book:	No

Course Instructor: Dr.Rabia Mansoor, Rizwan Ul Haq, Usman Ghous, Muhammad Yousaf Tahir Farooq

Student: Nam [illegible] Roll No. [illegible] Section. [illegible]

Instruction/Notes:
1. Verify at the start of the exam that you have a total of Four (4) question printed on Nine (9) pages including this title page.
2. The subjective part is required to be distributed after collecting the objective part.

<Subjective>

	Q-1 (objective)	Q-2	Q-3	Q-4	Q-5	Total
Total Marks	40	20	15	15	20	110
Marks Obtained		14	10	11.5	07	

Q2		14	20

Library Book Checkout System

You are assigned to create a library book checkout system in C++ utilizing loops to manage the library's collection of books. The program should provide users with the ability to view available books and perform book checkouts using iterative structures.

Requirements: (Must be present in your solution)

1. Implement a program that operates with a collection of 10 books, each identified by a unique ID (1 to 10).
2. Utilize arrays and loops to manage book information (IDs, titles, availability).
3. Write functions that allow users to:
4. Display available books.
5. Check out books by entering their IDs.
6. Display a menu using a loop to continuously prompt users for choices until they decide to exit.
7. Ensure appropriate error handling within loops for invalid inputs, such as non-existent book IDs or attempting to check out an already checked-out book.

Example Output:

```
Welcome to the Library Book Checkout System!
Menu:
1. Display available books
2. Check out a book
3. Exit

Enter your choice: 1
Available Books:
ID: 1, Title: "Book A", Available: Yes
ID: 2, Title: "Book B", Available: Yes

Enter your choice: 2
Enter the ID of the book you want to check out: 1
Book with ID 1 has been checked out successfully.

Enter your choice: 1
```

Available Books:
ID: 2, Title: "Book B", Available: Yes

Enter your choice: 3
Exiting the program. Thank you!

Objective:
Develop a book checkout system using C++ with an emphasis on using loops to iterate through book data and handle user interactions in a menu-driven environment for efficient book management within the library.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    bool exit = false;
    int choice, bookIDs = 1, userBookID;
    int availibilityBooks[10] = {0};
    char books[10][15] = { "Book A", "Book B", "Book C"
                         , "Book D", "Book E", "Book F"
                         , "Book G", "Book H", "Book I"
                         , "Book J" };

    (below declare, above while loop)
    while(exit != true)
    {
        cout << "Welcome to the Library checkout
                system!" << endl;
        cout << "Menu:" << endl;
        cout << "1. Display Available books." << endl;
        cout << "2. Check out a book." << endl;
        cout << "3. Exit." << endl;
        cout << "Enter your Choice:" << endl;
        cin >> choice;
        while(choice != 1 || choice != 2 || choice != 3)
        {
            cout << "Invalid input. Try again." << endl;
            cin >> choice;
        }
```

```
switch(choice)
{
    case 1:
    {
        for(int i=0; i<10; i++)
        {
            if(availibilityBooks[i] == 0)
            {
                cout<<"Available books: ";
                cout<<"ID: "<<bookIDs+1<<
                ", Title: "<<books[i]<<endl;
            }
        } else
    }
    case 2:
    {
        cout<<"Enter the ID of the book you
              want to checkout"<<endl;
        cin>>userBookID;
        cout<<"Book with ID: "<<userBookID
            <<"has been checked out successfully."
            <<endl;
        availibilityBooks[userBookID]++;
    }
    case 3:
    {
        cout<<"Thank You!"<<endl;
        exit = true;
    }
}
system("pause");
return 0;
```

Q3		15

a. Write down a program that will declare three two dimensional arrays of size N x M.

- These dimensions must be names constants and initialized to 3 and 4 respectively. **1**
- The code will initialize the first array with its $(row + col)^2$ for each location. **2**
- Then it will take input in second array from the user. **1.5**
- Initialize the third array with all 0 at the declaration time. (The initialization of the array should handle any size of N and M). **1.5**
- Then the code will update the third array with the difference of first array and second array index by index. The rows will be filled in reverse order. i.e. Array1[0][0] – Array2[0][0] is stored in Array3[0][3] **3**
- Print all three arrays **1**

Note: Do not use any hard coding. The written code must work for all valid values of N and M. Sample arrays after the execution of the code are shown below.

Array1

0	1	4	9
1	4	9	16
4	9	16	25

Array2

2	1	9	3
0	5	6	20
6	7	11	22

Array3

6	-5	0	-2
-4	3	-1	0
3	5	2	-2

b. Write a C++ program to check whether a given string is a subsequence of another given string. A subsequence is a sequence that can be derived from another sequence by removing some elements without changing the order of the remaining elements. **5**

Note: You are not supposed to use any built-in function to determine the subsequence. Both word1 and word2 must be char arrays of size 100 maximum. You can also take assumption that you must always determine if word2 is a subsequence of word1 or not.

Example1:
Input: word1 = 'apple'; word2 = 'apl';
Output: Yes, word2 is a subsequence of word1.
Example2:
Input: word1 = 'apple'; word2 = 'pla'
Output: No, word2 is not a subsequence of word1.

```
#include <iostream>
using namespace std;
int main()
{
    const int rowSize = 3;
    const int colSize = 4;
    int array1[rowSize][colSize], array2[rowSize][colSize]
        , array3[rowSize][colSize] = {0};
for(int i=0; i< rowSize ;i++)
{
    for(int x = 0; x < colSize ; x++)
    {
```

```
            array1[i][x] = (i+x) * (i+x);
        }
    }
    cout << " enter the numbers for second array " << endl;
    for( int i=0 ; i< rowSize ; i++)
    {
        for( int x=0 ; x< colSize ; x++)
        {
            cin >> array2[i][x];
        }
    }
    for( int i= 0, int j=0 ; i < rowSize ; i++, j++)
    {
        for( int x= colSize-1, int z=0 ; x>=0 ; x--, z++)
        {
            array3[i][x] = array1[j][z] - array2[j][z];
        }
    }
    for( int i=0 ; i< 3; i++)
    {
        for ( int x=0; x<3 ;x++)
        {
            for ( int j=0; j<4 ; j++)
            {
                if (i=0)
                {
                    cout << array1[x][j] << " ";
                }
                if (i=1)
                    cout << array2[x][j] << " ";
                if (i=2)
                    cout << array3[x][j] << " ";
            }
            cout << endl;
        }
    }
```

Q4		11.5	15

1. Consider the two code snippets given below. Fill the empty code skeleton for each of the code snippets without using the logical operators (&& and ||). (Points: 3+3 = 6)

Code Snippet 1

```
if (a > b || ! b >= c || a != c && c <= b)
        //print something 1
else
        // print something 2
```

Your answer:

```
if( a>b ✓ 0.5 ){

 cout<<"statement 1\n";
}
else if( !b>=c ✓ 0.5 )
{

 cout<<"statement 2\n";
}
else if( a!=c ){

 cout<<"statement 3\n";
}
else {

 cout<<"statement 3\n";
}
```

Code Snippet 2

```
if (a > b || ! b >= c && a != c || c <= b)
        //print something 1
else
        // print something 2
```

2.5

Your answer:

```
if( a>b ✓ 0.5 ){

 cout<<"statement 1\n";
}
else if( a!=c ){

 cout<<"statement 2\n";
}
else if( c<=b ✓ 0.5 ){

 cout<<"statement 3\n";
}
else {

 cout<<"statement 3\n";
}
```

Write a C++ program that will output a random word of length 4 using upper- and lower-case English alphabets (i.e., A to Z, a to z), for example: sWRT, PiCk, telQ. Use switch structure for selection wherever required. You cannot create a char variable/array for any purpose.

0476

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
    srand(time(0));
    int characterASCII1, characterASCII2, random;
    for(int i=0 ; i<4 ; i++)
    {
        characterASCII1=rand()%(90-65+1)+65;
        characterASCII2=rand()%(122-97+1)+97;
        random = rand()%2+1;
        switch(random)
        {
            case 1:
                cout << static_cast<char>(characterASCII1);
            case 2;
                cout<<static_cast<char>(characterASCII2);
        }
    }
    system("pause");
    return 0;
}
```

09

3+4=7

047

Q5 (a)		20

Write the **function definition** and proper **function call** statements with necessary variables declaration. No need of user inputs and to write any #include, library statements.

Write a C++ function named **validateMatrix**. The function should take **four** parameters: **two 2D integer arrays** (matrix A and matrix B) and their **dimensions** (rows and columns). The function should Check Consecutive Elements as follows; For each element in matrix A, check if the middle element is greater than both its left and right neighbors. If this condition holds, mark the corresponding element in matrix B with a special value **-1**. Ensure that the array indexes don't go out of bounds for the matrix.

Function Definition with complete body and logics:

```
int validateMatrix(int arr1[][cols],
    int arr2[][cols], int rows,
    int cols)
{
  for(int i=0; i<rows; i++)
  {
    for(int x=0; x<cols; x++)
    {
      if(arr1[i/2][x/2] > arr1[i/2+1][x/2+1] &&
        arr1[i/2-1][x/2-1])
          arr2[i/2][x/2] = -1;
}}}
```

02

Modify the function prototype of **validateMatrix function** such that user cannot modify the values of matrix A inside function body?

Updated prototype

```
int validateMatrix(int arr1[][cols],
  int arr2[][cols], static int rows, static int cols);
```

In **validateMatrix** function, make both 2D arrays as parameters with default values. Updated prototype?

Updated prototype

```
int validateMatrix(int arr1[][cols],
  int arr2[][cols], int rows, int cols);
```

Overload the **validateMatrix** function for bool, float and char datatypes as well? Just mention prototypes.

Updated prototype

What-if you're asked to pass above both 2D arrays as pass-by-reference parameters in **validateMatrix** function? Write Updated prototypes?

Updated prototype

Prototype :-

```
int validateMatrix(int arr1[][cols],
int arr2[][cols]);
```

- arrays are always passed as reference so we don't have to do anything.

The programing teacher at your university needs help in marking a True/False quiz. There are a total of 20 questions and no. of students are 50 in the class. The student's Roll_Nos along with their quiz answers are stored in a file **"submission.txt"**, where original solution of the quiz is stored in a separate file **"solution.txt"**. Every entry in submission file is the student Roll_No, followed by a blank, followed by the student's responses. For example, the first entry of submission file indicates that F230503 is the student's Roll_No, and answer to the question#01 is True, answer to the question#02 is False. Student did not attempt question#03 that's why its showing empty space and so on. Each correct answer contains 1 mark and no deduction for wrong answers. Write C++ code that reads the mentioned data from the below files, stores them in the program and outputs the student's Roll_No along with test marks in another file "marks.txt".

submissions.txt

F230503	TF TTFFTFTFFTTTFFTFF
F230712	FFFFFFFTFTFFTTTFFTFF
F238083	TTTTTTTTFFTTTFFTFF

solution.txt

TFFTFFTTTFFTFTFTFTT

marks.txt

F230503	5
F230712	10
F238083	12

Solution

All necessary variables declaration and libraries should be placed here.	`#include<iostream>`
	`#include<fstream>`
	`#include<cstring>`
	`using namespace std;`
		`char answers[20];`
		`char solution[20];`
		`int marks=0;`
		`char rollNum[8];`

File Reading code	`ifstream readFile;`
	`readFile.open("submission.txt");`
	`if(readFile.is_open())`
	`// reads file`
	`else`
	`cout << "couldn't open file" <<endl;`
		`while(readFile.eof == false)`
		`// read till end of file.`
		`readFile.close();`

Logical Code	`if(readFile.open is_open())`
	`{`
	`readFile>> solution;`
	`while(readFile.eof == false)`
	`{`
	`for(int i=0; i<20;i++)`
	`{ if(answers[i] == solution[i])`
	`marks++;`
		`else`
		`continue;`

File Writing Code	`ofstream writeFile;`
	`writeFile.open("marks.txt");`
	`if(writeFile.open is_open())`
	`// writes in File`
	`else cout << "couldn't open file" <<endl;`
		`writeFile.close();`

0262

Course Name:	Programming Fundamentals	Course Code:	CS1002
Degree Program:	BS(CS) | BS(SE) | BS(AI)	Semester:	Fall 2023
Exam Duration:	40 Min	Total Marks:	40
Paper Date:	Monday, December 18, 2023	Obtained Marks	
Sections:	ALL	No of Page(s):	3
Exam Term & Type:	Final Term | Closed Book	Required Answer Book:	No
Course Instructor	Dr.Rabia Maqsood, Rizwan Ul Haq, Usman Ghous, Muhammad Yousaf, Tahir Farooq		

Student: Name: _____ Roll No _____ Sect _____

Instruction/Notes: Verify at the start of the exam that you have a total of Two (1) question printed on Three (3) pages including this title page.

<Objective>

	Q-1	Total
Total Marks	40	40
Marks Obtained		38

Q1 Assuming there are no errors in the codes and all the headers are appropriately included, write the output of the given codes in the space provided. **8*5**

a.

```
int operate (int a, int b)
    {
        return (a * b);
    }
    float operate (float a, float b)
    {
        return (a / b);
    }
    int main ()
    {
        int x = 6, y = 7;
        float s = 4.0, u = 2.0;
        cout << operate (x, y);
        cout << operate (s, u);
        return 0;
    }
```

Answer:

4220.0

4/

b.

```
int addition(int x, int y, int z)
{
      int d;
      d = x + y + z;
      cout<<d;
      return d;
}
    int main( )
{
      int a,b,c,sum;
      a=10; b= 5; c=7;
      sum=addition(a , b , c);
      cout<<endl<<"Sum is = "<<sum;
}
```

Answer:

22
Sum is = 22

5/

c.

```
void function()
{
      int count=0;
      for( ;  ;  count++)
      {
            if(count<4)
                  cout<<count;
            else
                  break;
      }
}
int main()
{
      cout<<"Hi"<<endl;
      function();
      cout<<"FAST";
}
```

Hi
0123FAST

5/

d.

```
int main()
{
      for (int i = 0; i < 3; i++)
      {
            for (int j = 0; j <= i; j++)
            {
                  cout << "* ";
            }
            cout << endl;
      }
      for (int i = 0; i < 3; i++)
      {
            for (int j = i; j < 3; j++)
            {
                  cout << "* ";
            }
            cout << endl;
      }
}
```

5/

e.

```
int main()
{
int r,a=10,b=13,c=70;
r = (a>b?(a>c?5:7):(b>c?9:10));
cout<<r;
return 0;
}
```

10

5/

```
int main()
{
      int x = 10;
      while (x != 0)
      {
            x--;
            cout << x << endl;
            x--;
            if (x < 0)
            {
                  x = 0;
            }
      }
}
```

9
7
5
3
1

5/

```
int main()
{
      int num = 100;
      if(num > 1)
      if(num > 10)
      if(num > 100)
      cout << "3digit";
      else
      cout << "2digit";
      else
      cout << "1digit";
}
```

2digit

5

026

g

```
int number[6]={5,10,8,2,7,9};
  for ( int i=0; i<6-1; i++ )
  {
      for ( int j=0; j<6-1; j++ )
      {
        if ( number[ j ] < number[ j+1 ] )
        {
          int temp = number[ j ];
          number[ j ] = number[ j+1 ];
          number[ j+1 ] = temp;
        }
      }
   }
   for ( int i=0; i<6-1; i++ )
      {
      cout<<number[i]<<" ";
    }
```

10 9 8 7 5 2

4,

h