Student ID:.................................

Machine Number: ....

Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 1, Semester 1 (2023)

IT1010–Introduction to Programming
Session 3B

Duration: 3 Hours

November 2023

Instructions to Candidates:

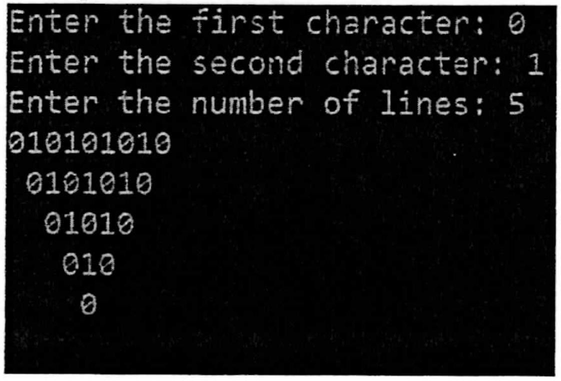
- ♦ This paper has 4 questions. Attempt all four questions.
- ♦ The total mark for the paper is 100.
- ♦ This paper contains 10 pages, including the cover page.
- ♦ Save all the programs in the given folder on your desktop with the given file names.
- ♦ Include your IT number in all your programs.
- ♦ DO NOT TAKE THIS PAPER FROM THE EXAMINATION HALL

Question 1 (20 marks)

Write a C program to input any two characters and a number (number of lines) from the keyboard to display an inverted triangle as shown in the examples below. The two characters should fill the inverted triangle in every other place.

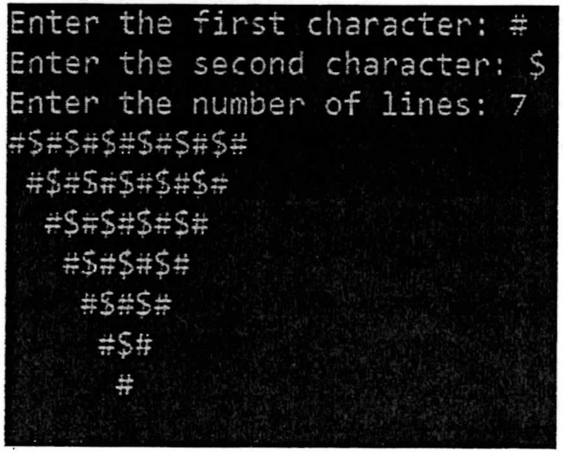
Sample Output

Example 1:



```
Enter the first character: 0
Enter the second character: 1
Enter the number of lines: 5
010101010
 0101010
  01010
   010
    0
```

Example 2:



```
Enter the first character: #
Enter the second character: $
Enter the number of lines: 7
#$#$#$#$#$#$#
 #$#$#$#$#$#
  #$#$#$#$#
   #$#$#$#
    #$#$#
     #$#
      #
```

Save your program as **3BQ1.c**

Question 2

(30 marks)

Part 1

Write a C program to perform the tasks related to 1-D arrays.

a) Create three 1-D integer arrays **array1**, **array2**, and **array3** with the lengths of 10, 5, and 2 respectively, and initialize all the arrays to zero.

Use repetition statements to perform the tasks in parts b, c, and d.

b) Set the following data to **array1** through the keyboard.

88	43	67	108	10	120	141	99	200	71

You must ensure that your program asks the values of array1 elements. You must enter as shown below.

```
Insert array1[0]:88
Insert array1[1]:43
Insert array1[2]:67
......................
......................
```

c) Find all the odd numbers in **array1** and store those numbers in **array2**.

d) Store the sum of the elements of **array1** and **array2** in **array3[0]**, **array3[1]** respectively.

e) Display the sum of odd numbers and even numbers according to the following format.

```
The sum of odd numbers: 421
The sum of even numbers: 526
```

Save your program as **3BQ2a.c**

Part 2

A Bank uses a 2D array called **transDetails** to store their customer transaction details. The array has three columns containing the following details.

The first column stores the total deposits – the total amount of money deposited to the account.
The second column stores the total withdrawals – the total amount of money withdrawn from the account.
The third column stores the withdrawal limit – the amount of money that can be withdrawn once.

Write a C program to do the following.

a) Declare a 2D array to store the details of 3 customers.
b) Input the total deposit amount, total withdrawal amount, and withdrawal limit of 3 customers and store in the array.
c) Display the details of 3 customers in the tabular format.
d) Ask the user to input the customer number (1/2/3), deposit or withdrawal (1- deposit, 2 – withdrawal), and amount. Update the array accordingly.
e) Display the details of all three (03) customers again after completing the transaction in part d)

<u>Sample Output</u>

```
Input the account details of customer 1
Input the total deposits:  10000.00
Input the total withdrawals: 5000.00
Input the withdrawal limit: 2000.00

Input the account details of customer 2
Input the total deposits:  12000.00
Input the total withdrawals: 5000.00
Input the withdrawal limit: -2000.00
Totals deposits, withdrawals, and withdrawal limits can't be negative.

Input the total deposits:  12000.00
Input the total withdrawals: 5000.00
Input the withdrawal limit: 3000.00

Input the account details of customer 3
.................................................................
...............................................................
```

10000.00	5000.00	2000.00
12000.00	5000.00	3000.00
4000.00	3000.00	2500.00

```
Select the customer (1/2/3): 3
Do you want to deposit or withdraw money (1 – deposit, 2 – withdraw) : 1
Amount: 4000.00
```

10000.00	5000.00	2000.00
12000.00	5000.00	3000.00
8000.00	3000.00	2500.00

Save your program as **3BQ2b.c**

00

Question 3 **(30 marks)**

"Sarasavi" Hardware shop sells different types of bricks for their customers. The details are given below.

Bricks Type	Description	Price of a brick (Rs.)
A	Small Bricks	40/=
B	Large Bricks	80/=
C	Cement Bricks	100/=

i) Write a function called `calcPayment()` to calculate and return the initial payment of a customer when the no. of bricks needed and brick type are passed as the function's parameters.

*Initial payment = No. of bricks needed * Price of a brick*

Function prototype is given below.

```
    float calcPayment(char brickType , int noBricks)
```

ii) The supplier will charge additional payment for the following services.

- If the customer needs transportation, 200/= will be charged for one kilometer.
- If the customer needs assistance with unloading, a payment of 2750/= will be charged.

Write a function called `calcAdditionalPayment()` to calculate and return the additional payment when the following are passed as the parameters.

- If the customer needs transportation, value Y will be passed to the function; otherwise, value **N** will be passed. The traveling distance in kilometers should be taken inside the function.
- If the customer needs assistance with unloading, value **Y** will be passed to the function; otherwise, value **N** will be passed.

Additional payment = Payment for transportation + *Payment for unloading*

The function prototype is given below.

```
float calcAdditionalPayment(char unloading , char transportation)
```

iii) Write a function called `testCalcPayment()`, which contains two assert statements to debug the implemented `calcPayment()` function.

iv) In your main function do the following,

a) Call `testCalcPayment()` function.

b) Allow the user to enter the number of bricks needed and the type from the keyboard. Call function `calcPayment()`

c) From the keyboard, allow the user to enter whether the customer needs transportation and whether the customer needs assistance with unloading. Call function `calcAdditionalPayment()`.

d) Calculate and display the net payment of a customer.

Net payment = Initial payment + Additional payment

e) Input the relevant details of three customers from the keyboard. Calculate and print the net payment of each customer.

Save your program as **3BQ3.c**

Question 4 **(20 marks)**

a) Write a C program to record the rainfall for one week. Enter the day of the week (ex: Sunday, Monday, etc.) and the rainfall(in mm) from the keyboard and store in a file called ***rain.txt***. Save your program as **3BQ4a.c**

Sample Input

Day	Rainfall (mm)
Sunday	70
Monday	85
Tuesday	98
Wednesday	55
Thursday	30
Friday	68
Saturday	100

b) Write a program to read the details from ***rain.txt.*** Find the day with the minimum amount of rainfall and display the result below.
Save your program as **3BQ4b.c**

Sample output

```
Thursday      30 mm
```

Grading Sheet

Question 1

Compile correctly	1.0
Execute correctly	
- Inputs	2.0
- Outputs	2.0
Printing the correct number of lines	3.0
Filling the correct number of spaces before the pattern starts	4.0
Filling tringle with the correct pattern	6.0
Coding conventions	2.0

Question 2 – Part A

Compile correctly	1.0
Execute correctly	
- Insert data into to array1 through keyboard	1.0
- Displaying the sum of even numbers and odd numbers	1.0
1D array	
- Create and initialize arrays	2.0
- Find even numbers in array1 and store those numbers in array2	5.0
- Store the sum of the elements of array1 and array2 in array3	3.0
- Display the sum of even numbers and odd numbers	1.0
Coding conventions	1.0

Question 2 – Part B

Compiles Correctly	0.5
Execute Correctly	
Input	2.0
Output	2.0
Correct array declaration	0.5
Insert values	1.0
Validate inputs	2.0
Perform a transaction and update the array	4.0
Display array	2.0
Coding Standards	1.0

Question 3:

Compile correctly	1.0
Execute correctly	
- Inputs	1.0
- Outputs	2.0
Implement function 1	6.0
Implement function 2	4.0
Implement test function	4.0
In main program	
- Take inputs	1.0
- Call functions in correct order	5.0
- Display output	2.0
- Correct use of repetition	2.0
Coding conventions	2.0

Question 4:

Compile correctly	1.0
Execute correctly	
- Data is available in the file	1.0
- Display Total Amount	2.0
Writing to file	
- Open file for writing	1.0
- Input several records from the keyboard	2.0
- Write them to the file	3.0
Reading from file	
- Open file for reading	1.0
- Read all the records	4.0
- Calculate the total amount	3.0
- Display the total amount	1.0
Proper coding standards are used	1.0