



Course Code: CS-1002	Course Name: Programming Fundamentals
Course Instructors: Mr. Muhammad Shahzad, Dr. Farooque Hassan Kumbhar, Dr. Abdul Aziz, Mr. Syed Zain Ul Hassan, Mr. Basit Ali, Ms. Sobia Iftikhar	
Open Date: October 15, 2023	Deadline:

Important instructions:

- You must never share your assignment with anyone. If any part of your assignment is found to be copied, then you will be awarded **0** marks as per the university policy.
- The assignment must be submitted before the deadline.
- Carefully read the submission instructions given below.

Submission:

- Create a separate folder for each of the tasks. You must write your solution by hand and then create its *pdf* file for submission.
- Create a main folder and rename it in the format “**k23-xxxx**” (replace **xxxx** it with your actual student ID). Then place all the separate folders for different tasks in this main folder.
- The main folder is to be submitted on Google Classroom.

Question: 01

[10 points]

You are given a 2D array that represents the batting performance of a cricket team. Each row in the array corresponds to a different batsman, and each column represents an inning played by that batsman. The values in the array represent the runs scored by each batsman in each inning.

Write a C program to perform the following tasks:

- 1) Input the number of batsmen and the number of innings from the user.
- 2) Input the batting performance for each batsman in each inning.
- 3) Calculate and display the following statistics for each batsman.
 - Total runs scored.
 - Average runs per inning
 - Highest score in a single inning
 - Number of centuries (innings with a score of 100 or more)
 - Number of half-centuries (innings with a score of 50-99)

Your program should display the statistics for each batsman individually.

Question: 02**[10 points]**

You are tasked with implementing a program to find the largest square submatrix of 1s in a given binary matrix. Write a C program that does the following:

1. Input a binary matrix (a 2D array of 0s and 1s) from the user. The matrix can be of any size.
2. Find and display the largest square submatrix of 1s in the input matrix.
3. Display the dimensions (rows and columns) of the largest square submatrix found.
4. If there are more than one largest square matrix, then find any one of them.

Question: 03**[20 points]**

You are assisting a traveler who is planning a trip and needs to book a flight for a specific day and time slot. The traveler is looking for the most convenient day and time slot where flights are available. You have access to a 2D array representing the flight availability for different days and time slots.

The 2D array below represents the availability of flights for different days of the week and two time slots: Morning and Evening. Each row corresponds to a day, and each column corresponds to a time slot. A value of 1 indicates that the flight is available (not full) for that time slot, and 0 indicates that the flight is full.

Day	Morning	Price	Evening	Price
Monday	1	300	0	-
Tuesday	1	320	1	310
Wednesday	0	-	1	280
Thursday	1	380	0	-
Friday	1	375	1	400

1. Analyze the flight availability for each day and time slot. Determine the day and time slot where the traveler has the highest chance of booking a flight on the basis user preferences i.e. based on time and price.
2. The traveler is flexible with the day of travel but prefers the morning time slot. Determine the day(s) when a flight is available in the morning and suggest the best option for booking based on this preference.
3. Considering the traveler's preference for an evening flight, identify the day(s) when an evening flight is available and suggest the best option for booking based on this preference.
4. The traveler is inquiring about flights on a specific day. Confirm if flights are available or not and provide the prices if available.

Question 04**[20 points]**

You are given a maze represented by a 2D array. The maze consists of walls represented by 'W', open paths represented by 'O', the starting point represented by 'S', and the exit represented by 'E'. The objective is to find a path from the starting point to the exit.

```
char maze[N][M] = {  
    {'S', 'O', 'O', 'W', 'W'},  
    {'O', 'W', 'O', 'O', 'W'},  
    {'O', 'O', 'O', 'W', 'O'},  
    {'W', 'W', 'O', 'W', 'O'},  
    {'W', 'W', 'O', 'E', 'W'}  
};
```

Write a C program to traverse the maze of 5 x 5 and find a path from the starting point to the exit.

Constraints

- There will be at least one path from the starting point to the exit. In case of more than one path then consider only one.
- You can visit only right and down direction.
- Output will be only indexes of correct path.

Input Original Maze:

S O O W W

O W O W W

W O O W O

W W O W O

W W O E W

Output Maze after traversal:

0,0 0,1 0,2 1,2 2,2 3,2 4,2 4,3

Question 05**[20 points]**

Ramanujan-Hardy Numbers 2-way are numbers that are the sum of two cubes two different ways.

Following are the first few Ramanujan-Hardy 2-way numbers:

1729, 4104, 13832, 20683, 32832, 39312, 40033, 46683, 64232, 65728, 110656, 110808, 134379,
149389, 165464, 171288, 195841, 216027, 216125, 262656, 314496, 320264, 327763, 373464, 402597,
439101, 443889, 513000, 513856, 515375, 525824, 558441, 593047, 684019, 704977

E.g., $1729 = 12^3 + 1^3 = 9^3 + 10^3$.

A. Write a C program to find all Ramanujan-Hardy numbers less than n^3 .

B. Draw a trace table on a paper for the input 1729.

Question 06

[10 points]

Given a 1D array of N integers, and a target integer t.

Write a 'C' program to find all different number pairs in the array that have sum equal to t.

Sample Input:

List: 1, 8, 10, 12, 7, 4, 3 and **t:** 11

Sample Output:

Pairs: (1, 10), (7,4), (8,3)

Question 07:

[10 points]

Your task is to developing a program to manage and sort shirt details for a clothing store. The shirt details include the age and price for each shirt. Write a C program that sorts the shirt details based on age in ascending order and, within the same age, sorts them based on price in descending order.

You are given 2D array : ages (representing the age of each shirt) and prices (representing the price of each shirt). Assume any number of shirts and its prices by yourself.

Your expected output will show two different sorted list i.e., Sorted list in ascending order with respect to **Age** and Sorted list in descending order with respect to **Price**.