FAST NUCES Peshawar Campus

Programming Fundamentals

Spring 2024, Section B

Assignment: 02

Deadline: 7th May, 2024

Late submissions will not be accepted. You are required to create a repository on your GitHub account and upload your assignment to Git by the specified date(Making a Git repo is a bonus marks). Additionally, on Google Classroom, you must upload a single zip file containing three files in the following format: **‘Q1-23P-000‘, ‘Q2-23P-000‘, ‘Q3-23P-000‘**. Your zip file should be named with your roll number, for instance, **‘23P-0000.zip‘**. Submit the form with the GitHub repository URL before the deadline.

Question 01

The Lost Temple of Angkor

Background

The Lost Temple of Angkor has been hidden for centuries, with stories of incredible treasures hidden deep within its ancient walls. Recently, explorers have unearthed a map that reveals a secret grid of numbers inscribed on the temple's stone floor. These numbers are believed to hold the key to finding the greatest treasure in the temple.

As an expert cryptographer, you have been tasked with deciphering the map to determine the location of the treasure. The map contains a 20x20 grid of numbers. Hidden within this grid are sequences of four adjacent numbers in various directions—horizontal, vertical, and diagonal. The key to unlocking the temple's secrets lies in finding the greatest product of four adjacent numbers.

Problem Statement

Given a 20x20 grid of numbers, your task is to find the largest product of four adjacent numbers in the same direction (horizontally, vertically, or diagonally). You must consider all possible sequences of four adjacent numbers and calculate their product. Your objective is to determine the maximum product among all possible sequences.

Input

The 20x20 grid is as follows:

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

Output

Your output should be the largest product of four adjacent numbers in the same direction within the 20x20 grid. The output should clearly state which direction yielded the maximum product (horizontal, vertical, diagonal).

Hints

Consider using 2D arrays to represent the grid. Develop logic to traverse the grid and calculate products in all possible directions. Keep track of the maximum product while iterating through the grid.

Deliverables

Submit the source code of your C program, with comments explaining the logic and key sections. Provide a brief explanation of the approach taken to solve the problem. Include any additional information that might help in understanding the problem and your solution.

The Longest Collatz Sequence

Question 02

Story

The Collatz Conjecture, also known as the "3n + 1" problem, is a famous unsolved problem in mathematics. The conjecture involves a sequence of numbers generated from a starting point following these rules:

1. If the number is even, divide it by 2.
2. If the number is odd, multiply it by 3 and add 1.

This process repeats until the number becomes 1. The Collatz Conjecture suggests that this process will always reach 1, regardless of the starting number. As a mathematician, you have been tasked with exploring the Collatz sequences to find the starting number under a given limit that produces the longest sequence.

Problem Statement

Given a limit N, your task is to find the starting number under N that produces the longest Collatz sequence. You must determine the length of each Collatz sequence and identify the starting number with the longest length. Write a C program that calculates the length of Collatz sequences for all starting numbers under N and returns the number with the longest sequence.

Input

- A single integer $N > 1$, representing the upper limit for the starting numbers.

Output

- The starting number under N that produces the longest Collatz sequence.
- The length of the longest Collatz sequence.

Hints

- Consider implementing a function that calculates the Collatz sequence length for a given starting number.
- Keep track of the lengths to identify the longest sequence.
- Use a loop to iterate through all starting numbers under N.

Example Scenario

Given a limit of 10, the longest Collatz sequence is produced by the starting number 9, with a length of 20.

Tasks

1. **Collatz Sequence Function**: Write a function that calculates the Collatz sequence length for a given starting number.
2. **Longest Collatz Sequence**: Implement a function that finds the starting number under N with the longest Collatz sequence.
3. **Main Program**: Create a program that takes an upper limit N and outputs the starting number with the longest Collatz sequence and its length.

Deliverables

- The source code of your program, with comments explaining the logic and key sections.
- A brief explanation of the approach taken to solve the problem.

Poker Hands Comparison

Question 03

Story

In a high-stakes poker tournament, two players are competing for the grand prize. To determine the winner, their poker hands must be compared according to standard poker rules. Each hand consists of five cards, and the hands can be ranked in a specific order, from the highest (Royal Flush) to the lowest (High Card).

The referee needs your help to decide which player wins the most games, given a series of poker hands for both players. Your task is to create a C program that compares poker hands and determines which player has the winning hand.

Problem Statement

Given a list of poker hands for two players, each with five cards, your task is to determine which player wins more hands. The poker hands should be evaluated according to standard poker rules. The program must compare the hands to identify the winner.

Write a C program that compares poker hands and outputs the player with the winning hand.

Input

A list of poker hands, where each line contains 10 cards separated by a space. The first five cards represent Player 1's hand, and the next five cards represent Player 2's hand.

Output

The player with the winning hand for each set of poker hands. If the hands are equal, output "It's a tie."

Hints

- Consider evaluating poker hands according to the standard hand rankings: Royal Flush, Straight Flush, Four of a Kind, Full House, Flush, Straight, Three of a Kind, Two Pairs, One Pair, and High Card.
- Implement a function to parse the input and convert it into card ranks and suits.
- Use loops to compare hands and determine the winner.
- Keep track of which player wins more hands.

Example Scenario

Consider the following five hands dealt to two players:

Hand	Player 1	Player 2	Winner
1	5H 5C 6S 7S KD	2C 3S 8S 8D TD	Player 2
2	5D 8C 9S JS AC	2C 5C 7D 8S QH	Player 1
3	2D 9C AS AH AC	3D 6D 7D TD QD	Player 2
4	4D 6S 9H QH QC	3D 6D 7H QD QS	Player 1
5	2H 2D 4C 4D 4S	3C 3D 3S 9S 9D	Player 1

Table 1: Poker Hands Comparison

Tasks

1. **Parse Poker Hands**: Implement a function to parse a line of input into two separate poker hands.
2. **Evaluate Poker Hands**: Create a function to evaluate the rank of each poker hand.
3. **Compare Poker Hands**: Implement a function to compare two poker hands and determine the winner.
4. **Main Program**: Create a program that reads a list of poker hands and outputs which player wins more games.

Deliverables

- The source code of your C program, with comments explaining the logic and key sections.
- A brief explanation of the approach taken to solve the problem.
- Test cases demonstrating the correctness of your program with different sets of poker hands.

Submission Guidelines

- Submit your code as a single C source file with appropriate comments.
- Provide a brief explanation of your approach and test cases demonstrating its correctness.
- Ensure your solution correctly compares poker hands and outputs the winner.