

Mid-Term Examination – 1

Computer Programming with C

MCA102

Duration: 1 hour

Answer any one

Total Marks: 30

The Adventure of Maya the Explorer

Once upon a time, in the land of Numbers, there lived a young explorer named Maya. Maya loved to find hidden treasures, and in her world, the treasures were the largest and smallest numbers hidden in a sea of other numbers. But Maya wanted to be smart about her adventures, so she decided to create a small program to help her find these treasures.

Maya decided to split her code into two parts: one part to declare the treasure-finding methods and another part to define what these methods do. This way, her treasure-finding tools would be clean and easy to use!

In **search.h**:

- Maya declares and defines two special treasure-hunting functions:
 - **findLargest**: A function that finds the biggest number (the largest treasure).
 - **findLargest** starts by assuming that the first number Maya finds is the largest treasure. As Maya travels through the array of numbers, if she finds a bigger number, she updates her discovery.
 - **findSmallest**: A function that finds the smallest number (the smallest hidden treasure).
 - **findSmallest** works similarly, assuming the first number is the smallest and updating her finding whenever a smaller number is discovered.

In **main.c**:

- Maya starts her journey by entering how many treasures (numbers) she wants to search through.
- She inputs these treasures (numbers) into an array.
- Maya uses the **findLargest** function to discover the biggest treasure and the **findSmallest** function to uncover the smallest treasure.
- The program then prints out the largest and smallest treasures Maya found.

How It Works (Maya's Complete Adventure)

1. **main.c** asks Maya to enter the number of elements (numbers) and then to input each element into an array.
2. Maya uses the functions **findLargest** and **findSmallest** (from **search.h**) to find the largest and smallest numbers, respectively.
3. The results are displayed: the largest and smallest numbers in the array.

By splitting the code into **search.h** (declarations & definitions) and **main.c** (usage), Maya organised her treasure-hunting tools neatly and made sure they were reusable for future adventures!

OR

The Story of Zara the Zookeeper

Zara is a zookeeper responsible for keeping track of how much food each animal eats daily. She has a list of animals and needs to:

1. Calculate the total food consumed.
2. Find out which animal eats the most.
3. Find the average amount of food consumed by all animals.

In **zoo.h**, Zara declares and defines three functions:

- **totalFood**: Calculates the total food eaten by all animals.
 - **totalFood**: Loops through the array and sums up the food consumed.
- **maxFood**: Finds the maximum amount of food eaten by a single animal.
 - **maxFood**: Loops through the array to find the animal that ate the most food.
- **averageFood**: Calculates the average food consumed by all animals.
 - **averageFood**: Calls **totalFood** and divides the result by the number of animals to get the average.

In **main.c**:

- Zara enters the number of animals and their corresponding food consumption into an array.
- The program calculates:
 - **Total food** consumed using **totalFood**.
 - **Maximum food** eaten by a single animal using **maxFood**.
 - **Average food** consumption using **averageFood**.
- Zara's daily report is then printed out.

How Zara's Program Works

1. Zara inputs the number of animals and their food consumption.
 2. The program calculates:
 - **Total food** eaten by all the animals.
 - **Maximum food** eaten by a single animal.
 - **Average food** consumed by all animals.
 3. The results are printed for Zara's report.
-