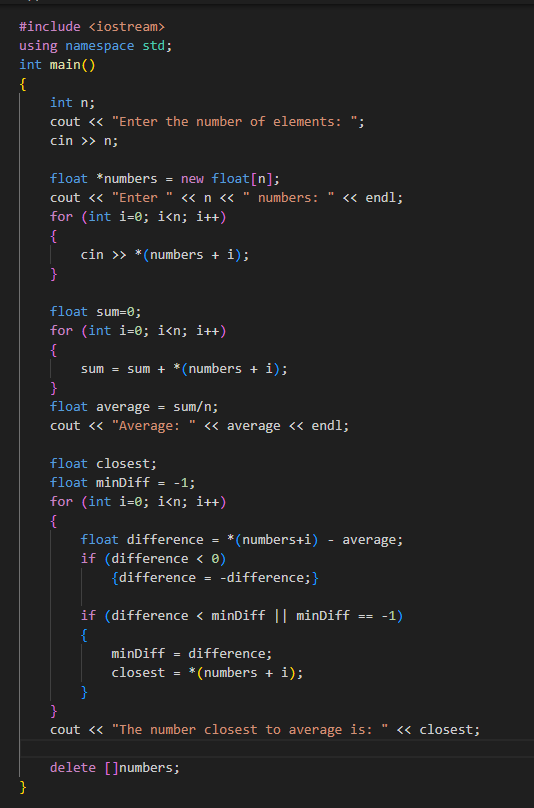
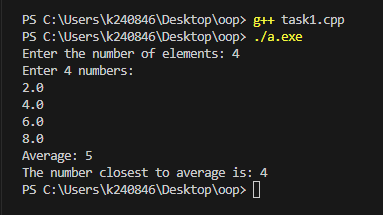
**OOP LAB 2**

**Task 1)** Write a C++ program that reads a group of n numbers from the user and stores them in a dynamically

allocated array of type float. Then, the program should:

* Calculate the average of the numbers.
* Find the number closest to the average.
* Print the average and the number closest to it.
* Use pointer notation wherever possible.

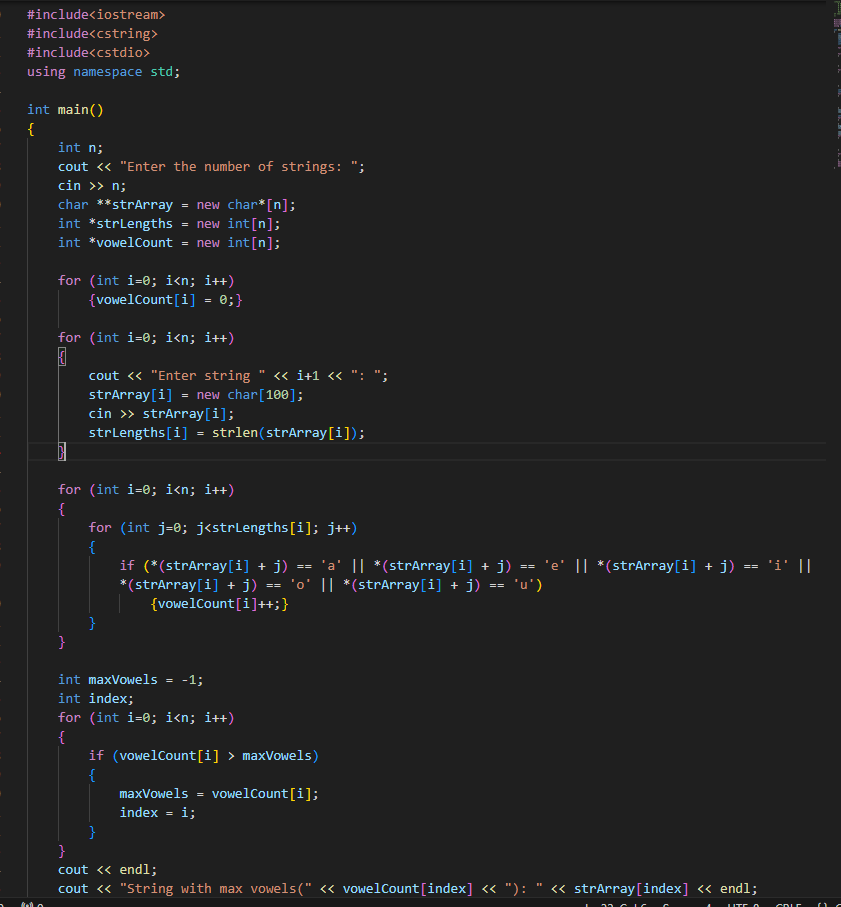


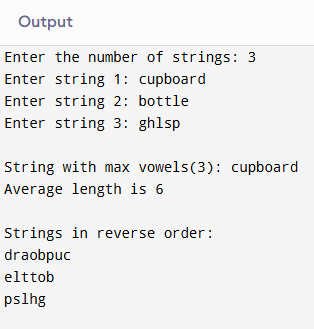
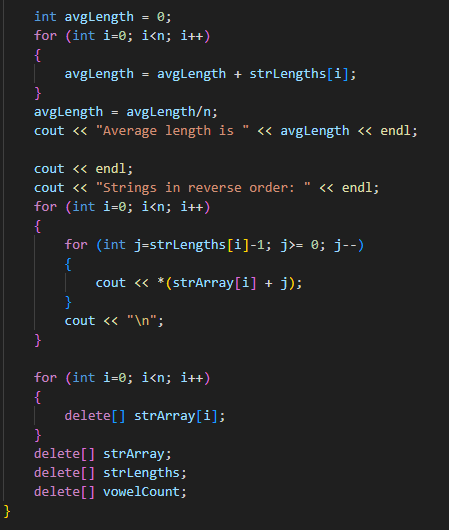


**Task 2)** Write a C++ program that:

* Reads n strings from the user and stores them in a dynamically allocated array of char\*.
* Prints the strings in reverse order using pointer arithmetic.
* Finds and prints the string with the most vowels (a, e, i, o, u).
* Calculates and prints the average length of all the strings.

Note: Use pointer notation wherever possible.

If there are multiple strings with the same number of vowels, print the first one encountered

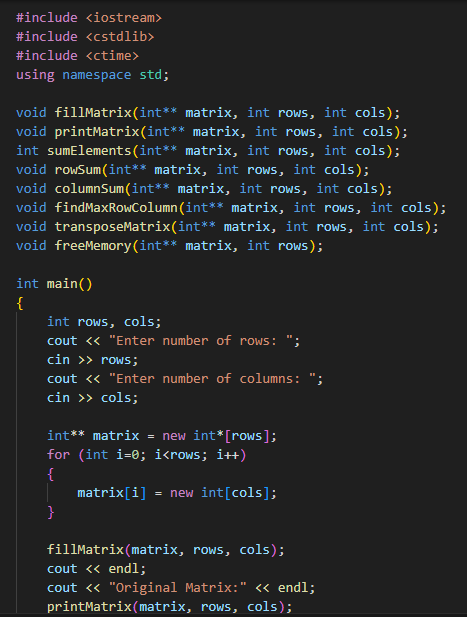


**Task 3)** Write a C++ program that:

* Dynamically allocates a 2D array using pointers (not using vector or standard containers).
* Fills the array with random integers between 1 and 100.
* Pass the 2D array to function to perform these tasks:
* Calculates and prints The sum of all elements in the array.
* Calculates and prints The sum of each row and each column.
* Calculates and prints The row and column with the highest sum.
* Pass the 2D array to a function to transpose the matrix and print the resulting matrix.

Free the dynamically allocated memory.

Note: Use functions to perform the calculations and matrix operations (do not write all code inside main() ). Handle edge cases, such as when the array has no elements or is improperly allocated.

**Sample Output:**

**Original Matrix:**

[ 12 35 56 ]

[ 8 45 67 ]

[ 23 54 34 ]

Sum of all elements: 434

Row sums: 103, 120, 111

Column sums: 43, 134, 157

Row with highest sum: Row 2

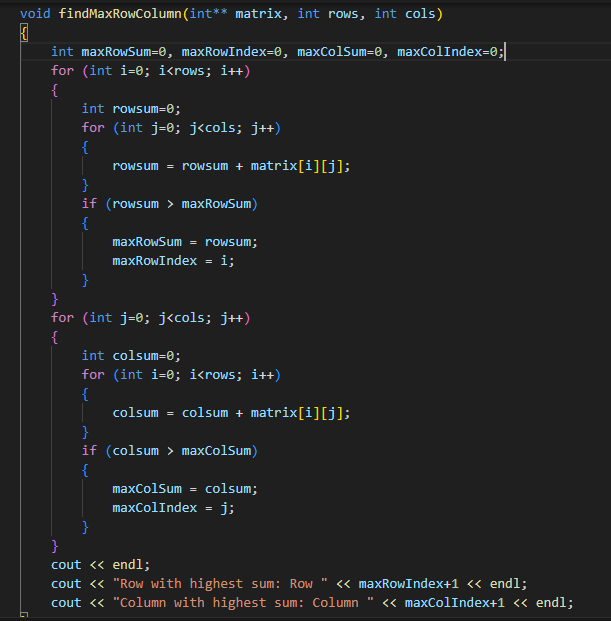
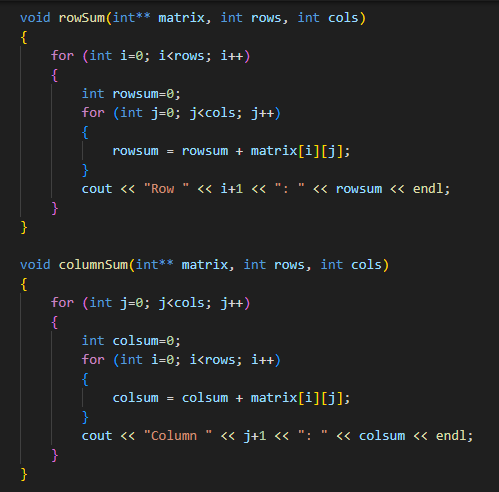
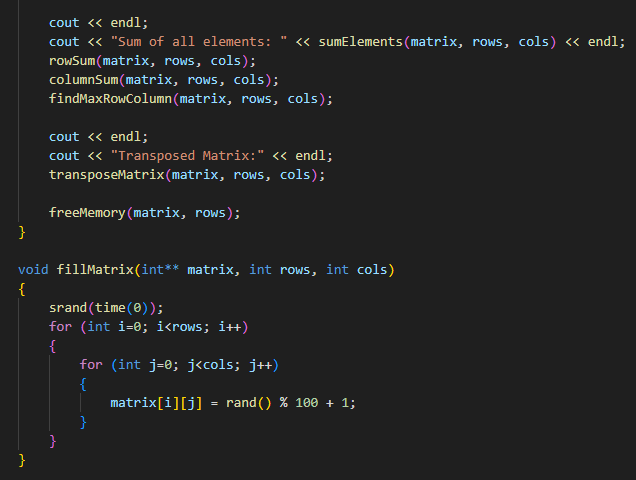
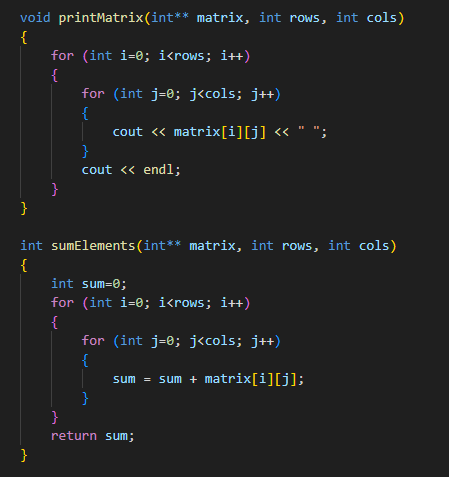
Column with highest sum: Column 3

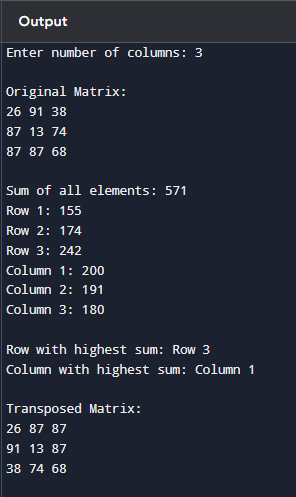
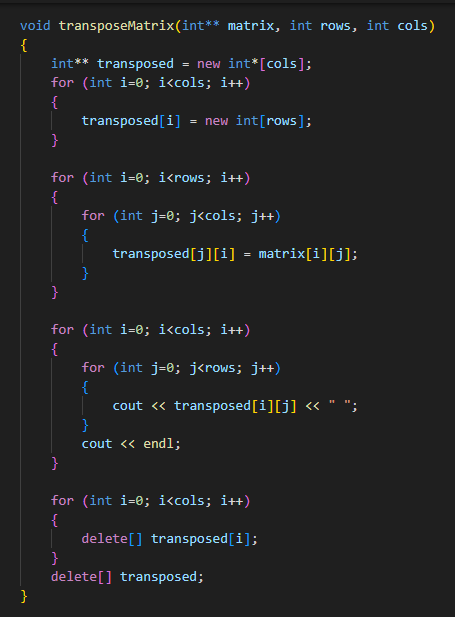
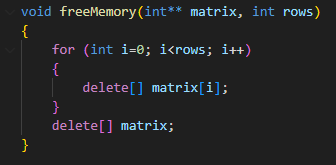
**Transposed Matrix:**

[ 12 8 23 ]

[ 35 45 54 ]

[ 56 67 34 ]





**Task 4)** You are required to write a C++ program that will creates a function named unique that will take array

as input. the array may contains the duplicates values but you have to process on the array and have to

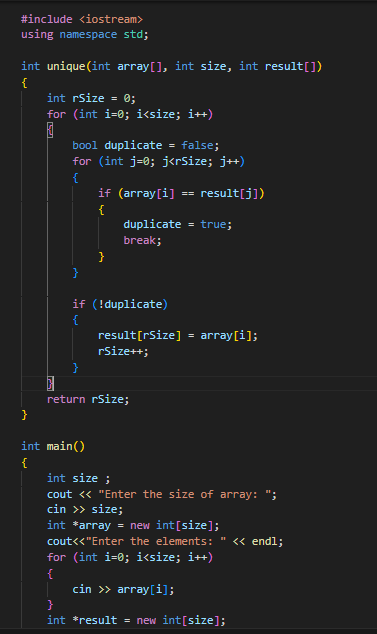
return the array which must contains only unique values not duplicates.

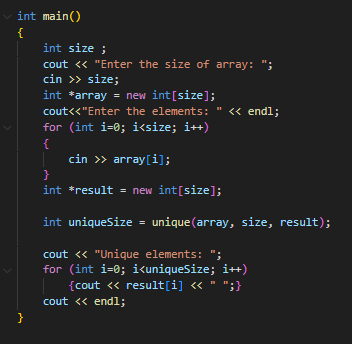
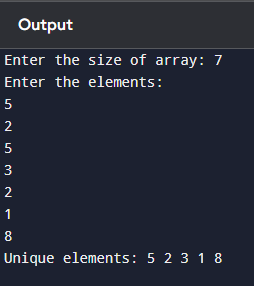
Example:

Size of array: 7

Elements of array: [5, 2, 5, 3, 2, 1, 8]

Unique elements: 5 2 3 1 8





**Task 5)** You are required to write a c++ function swap\_string that shifts the last n characters of a string to the

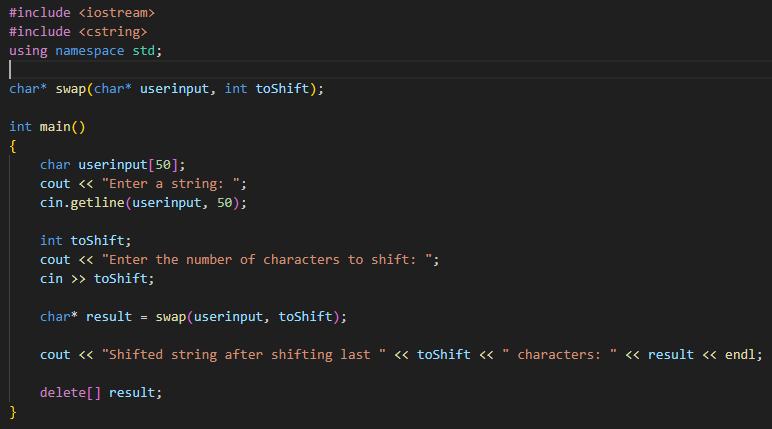
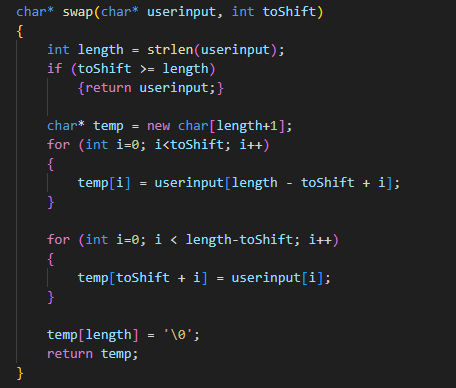
front n times. It will take str and int as parameters. And will return the new string after shifting.

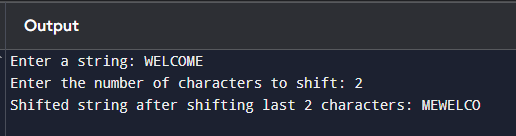
Note: You have to work with pointers.

Output:

Enter a string: WELCOME

Enter the number of characters to shift: 2

Shifted string after shifting last two characters: MEWELCO



**Task 6)** You are tasked with implementing a simple Student Registration System in C++. Define two structures,

Register and Student, where Register contains attributes courseId and courseName, and Student inherits

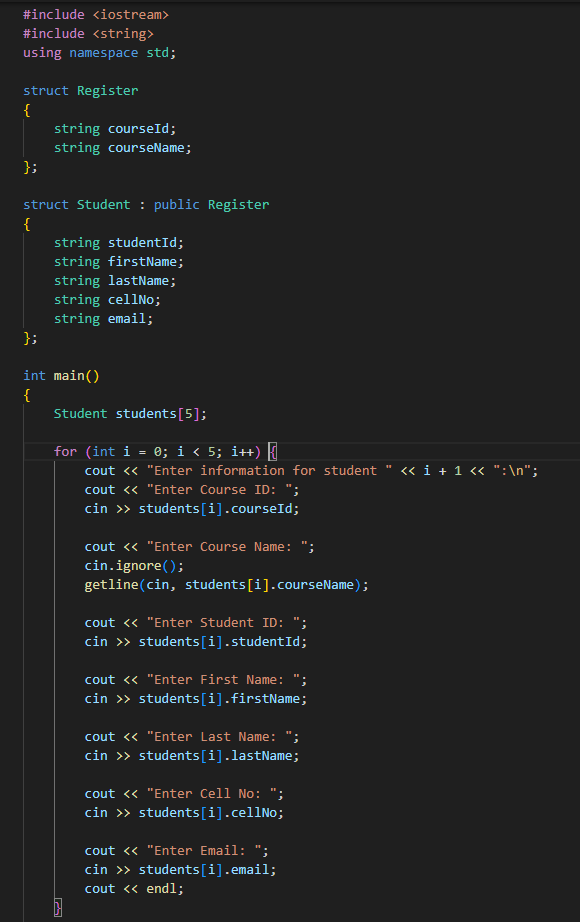
from Register while having additional attributes such as studentId, firstName, lastName, cellNo, and

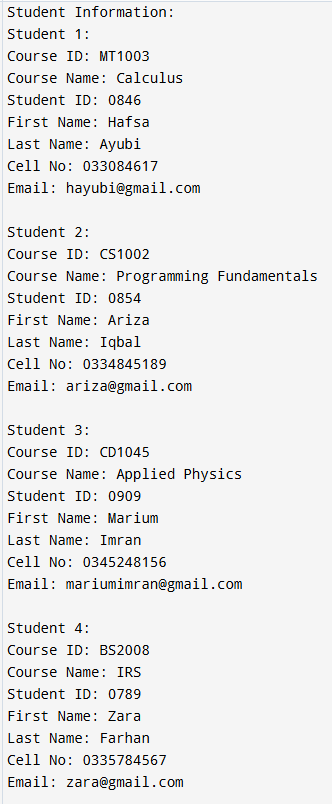
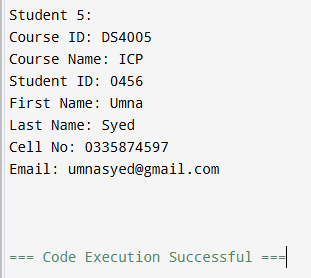
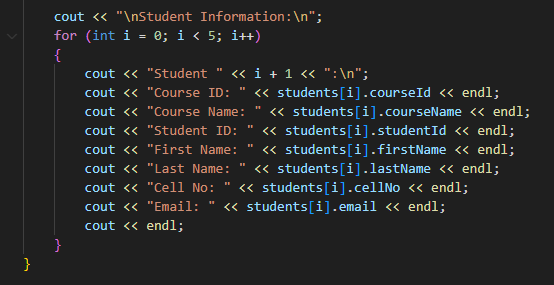
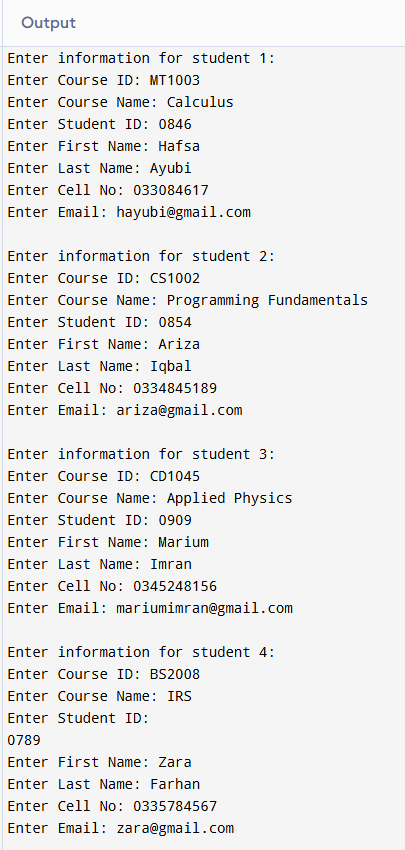
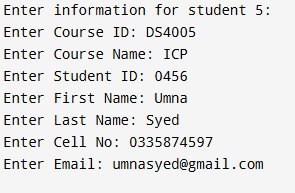
email. Your goal is to create an array of Student structures to store information for five students. Write a

C++ program that accomplishes the following tasks:

* Implement the Register and Student structures.
* Inherit the Register structure in the Student structure.
* Create an array of Student structures to store information for 5 students.
* Take input for each student, including their courseId, courseName, studentId, firstName,

lastName, cellNo, and email.

* Display the information for all 5 students.



**Task 7)** You are tasked with building a simple product management system for an online store.

1. Create a function that allows the addition of a new product to the system. The function should

take parameters such as product name, price, quantity, and any other relevant details.

2. Implement a function that takes a product ID as input and displays detailed information about the

product, including its name, price, quantity in stock, and any other relevant details.

3. Design a function that enables the update of product information. It should take a product ID as

well as the new details (e.g., updated price, quantity, etc.) and modify the existing product’s information accordingly.

4. Create a function that removes a product from the system based on its product ID. Ensure that

the inventory is updated after the removal.

