sequential search.

Q1. **[10 Marks]**

An online shopping platform stores the product IDs of items added to a customer's shopping cart in an array. The customer wants to check if a specific product (based on its ID) is already added to the cart. Write an Assembly program that takes a product ID as input from the user and performs a sequential search in the stored shopping cart data. If the product ID is found, the program should display the index of the product in the cart; otherwise, it should indicate that the product is not in the cart.

```
.data
arr WORD 0549h, 0558h, 0585h, 0596h, 0608h, 0628h, 0648h, 0656h, 0670h, 0678h
target WORD ?
msg1 BYTE "Product found at index: ", 0
msg2 BYTE "Product not found in the cart!", 0
msg3 BYTE "Enter the product ID to find: ", 0
```

SAMPLE INPUT-OUTPUT:

```
Enter the product ID to find: 0596
Product found at index: 3

Enter the product ID to find: 0610
Product not found in the cart!
```

CLO # 2: Develop the ability to write assembly language programs utilizing arrays, indexed addressing, and rotation instructions (ROL/ROR) to perform dynamic data manipulatio..

Q2. **[10 Marks]**

Take a word array with the size of your own actual name (full name) stored in it then rotate it till your first name becomes last. Your function should be dynamic not hard coded, such that if i replace the name with any other name, it works fine.

Actual Name (before Rotation)

A	R	H	A	M		R	A	S	H	E	E	D

Name (After Rotation)

R	A	S	H	E	E	D		A	R	H	A	M

CLO # 3: Develop the ability to write assembly language programs using jmp, cmp, loops, and arrays for decision-making, iteration, and data manipulation. Gain proficiency in string handling and modular programming through tasks like string reversal, case conversion, and prime number checking.

Q3. **[7.5 + 7.5 Marks]**

(a) Write a procedure to reverse and update the words in a string. It receives the string through a stack and modifies the string by the following pattern. Here is an example:

INPUT STRING:
Brazil have won the FIFA World Cup five times, the most in history, while Germany and Italy have won it four times each.

MODIFIED STRING:
Each times four it won have Italy and Germany while, history in most the, times five Cup World FIFA the won have Brazil.

(b) Write a procedure named **CountEvenNumbers** that takes an array of 32-bit integers and its size as input. The procedure should count how many even numbers are present in the array and store the result in the EAX register.
Create a test program that does the following:

Prompts the user to enter a sequence of integers.
Calls **CountEvenNumbers** to count the even numbers.
Displays the count of even numbers.
Continues until the user enters -1.

```
Sample Output:

Enter integers separated by spaces (end with -1): 2 5 8 11 -1
You entered 2 even numbers.

Enter integers separated by spaces (end with -1): 1 3 7 9 13 -1
You entered 0 even numbers.

Enter integers separated by spaces (end with -1): 10 20 30 40 -1
You entered 4 even numbers.

Exiting...
```

CLO # 3: Develop the ability to write assembly language programs using jmp, cmp, loops, and arrays for decision-making and data manipulation. Apply inline and external assembly to solve mathematical problems and translate high-level code.

Q4. **[15 Marks]**

Write an equivalent assembly language code for the following High Level Language code:

```
int main() {
    int arr[10], freq[10] = {0};
    int mostFrequentElement, maxFreq = 0;

    // Input 10 elements
    printf("Enter 10 elements of the array:\n");
    for (int i = 0; i < 10; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    // Count frequency of each element
    for (int i = 0; i < 10; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (arr[i] == arr[j]) {
                freq[i]++;
            }
        }
    }

    // Find the most frequent element
    for (int i = 0; i < 10; i++) {
        if (freq[i] > maxFreq) {
            maxFreq = freq[i];
            mostFrequentElement = arr[i];
        } else if (freq[i] == maxFreq && arr[i] < mostFrequentElement) {
            mostFrequentElement = arr[i];
        }
    }

    // Display result
    printf("\nMost frequent element is: %d\n", mostFrequentElement);

    return 0;
}
```

SAMPLE INPUT-OUTPUT:

```
Enter 10 elements of the array:
Enter element 1: 5
Enter element 2: 7
Enter element 3: 5
Enter element 4: 3
Enter element 5: 3
Enter element 6: 3
Enter element 7: 8
Enter element 8: 7
Enter element 9: 5
Enter element 10: 7

Most frequent element is: 5
```

Good Luck!