Data Structures and Algorithms I

[Tutorial 01] - Index-225036C

One Dimensional Array

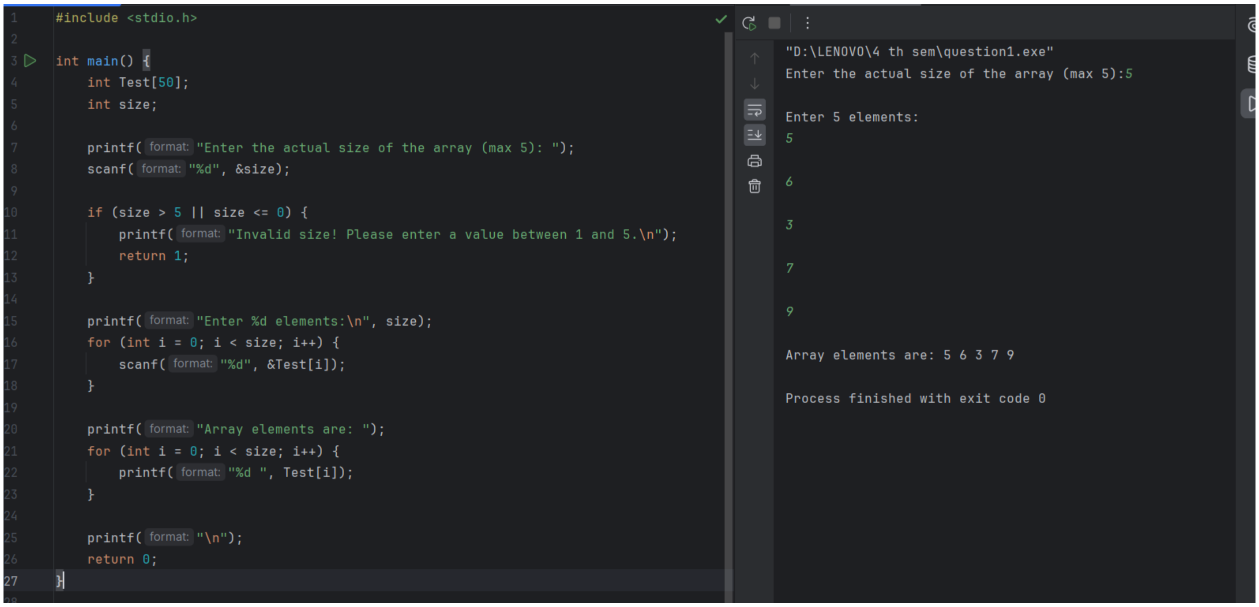
1). Write a C program to take inputs from user and print the array values using for loops. (For one-D Array)

Hint: use array name as **Test** and actual size of array is 5 and array maximum size is 50.

Use the array values as 5, 6,3,7,9

Write a required code snippet for asking user what is the actual size of array.

Hint: use new variable **size**

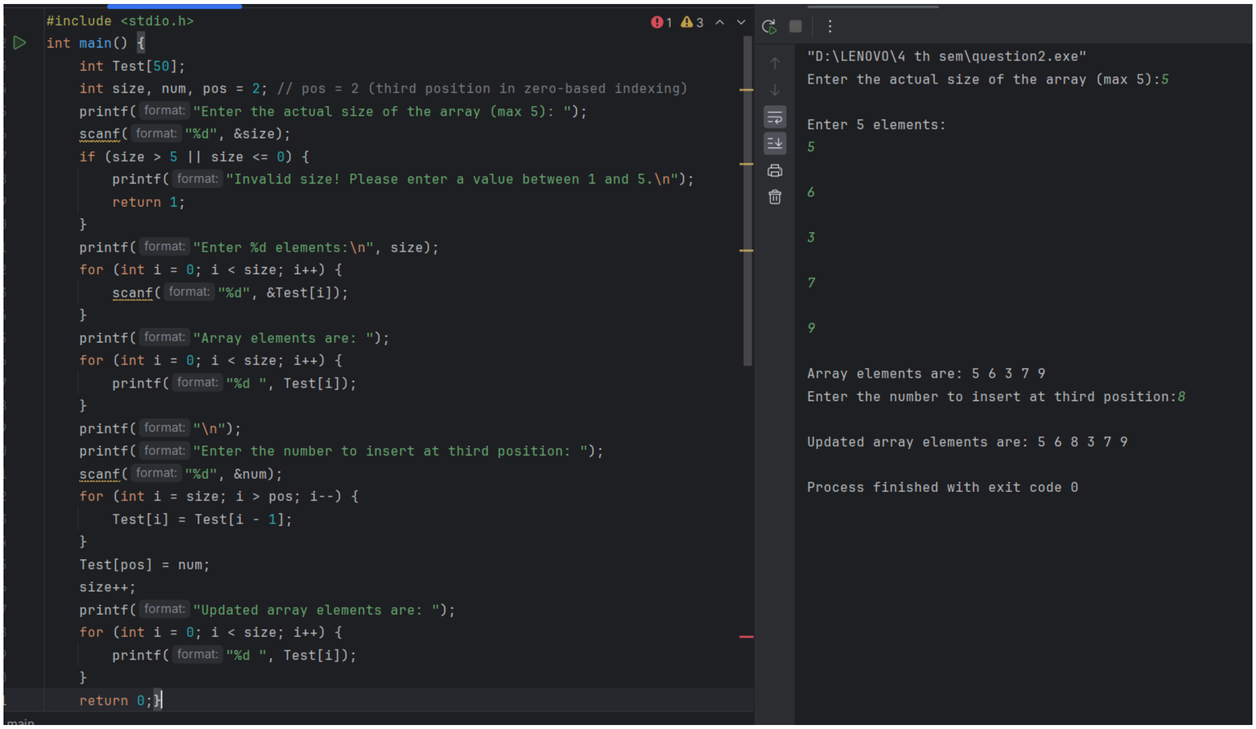


```c
#include <stdio.h>

int main() {
    int Test[50];
    int size;

    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);

    if (size > 5 || size <= 0) {
        printf("Invalid size! Please enter a value between 1 and 5.\n");
        return 1;
    }

    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }

    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }

    printf("\n");
    return 0;
}
```

```
"D:\LENOVO\4 th sem\question1.exe"
Enter the actual size of the array (max 5):5

Enter 5 elements:
5


6


3


7


9


Array elements are: 5 6 3 7 9

Process finished with exit code 0
```

2). Update the above program to insert an element to third position of the above array. (Use swapping method; start to swap from last index).

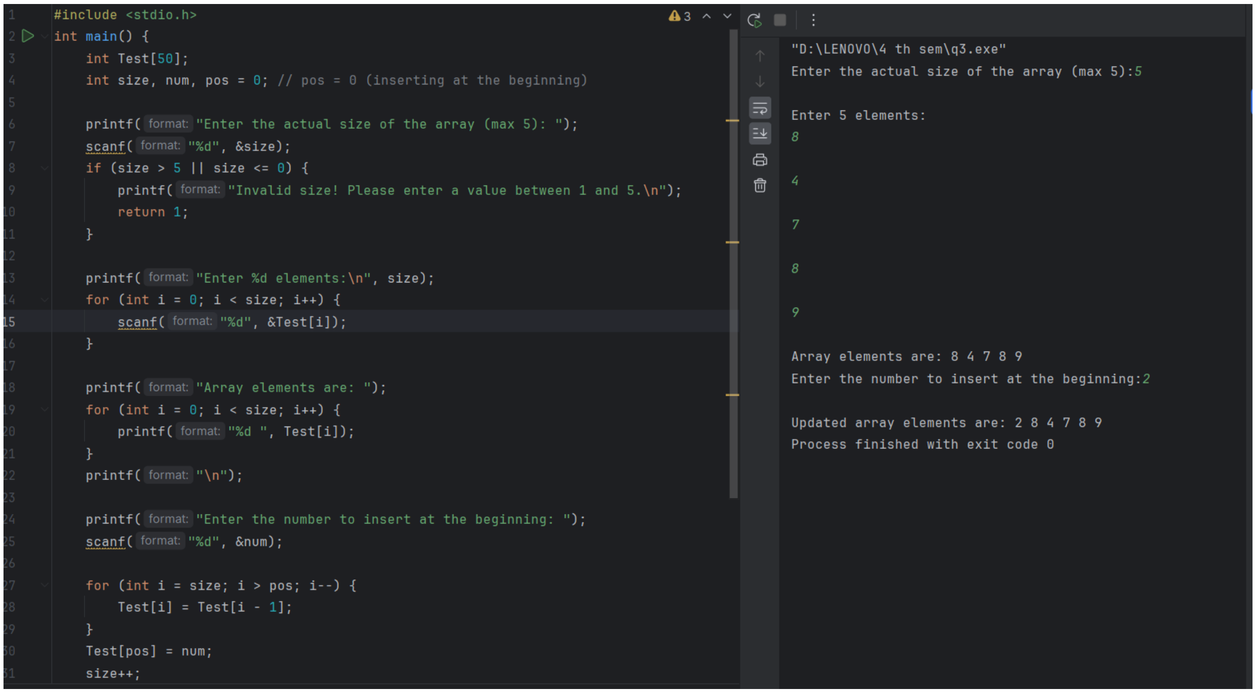
Hint: use two new variables as **num** and **pos**. Write the code segment to read the values for these two variables from the user.



```c
#include <stdio.h>
int main() {
    int Test[50];
    int size, num, pos = 2; // pos = 2 (third position in zero-based indexing)
    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);
    if (size > 5 || size <= 0) {
        printf("Invalid size! Please enter a value between 1 and 5.\n");
        return 1;
    }
    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }
    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    printf("\n");
    printf("Enter the number to insert at third position: ");
    scanf("%d", &num);
    for (int i = size; i > pos; i--) {
        Test[i] = Test[i - 1];
    }
    Test[pos] = num;
    size++;
    printf("Updated array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    return 0;}
```

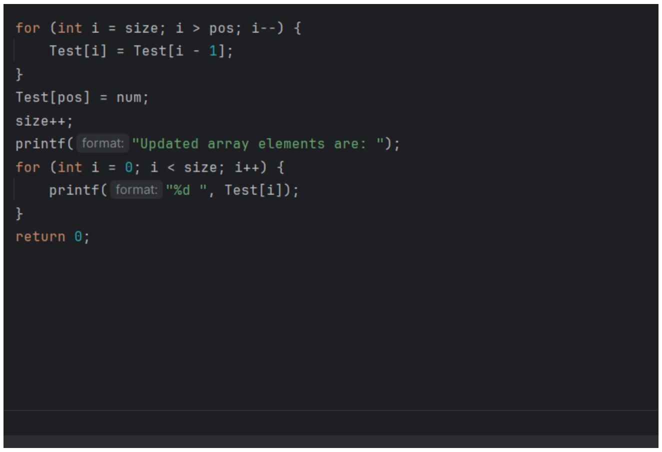
```
"D:\LENOVO\4 th sem\question2.exe"
Enter the actual size of the array (max 5):5

Enter 5 elements:
5

6

3

7

9

Array elements are: 5 6 3 7 9
Enter the number to insert at third position:8

Updated array elements are: 5 6 8 3 7 9

Process finished with exit code 0
```

3)Update the program to insert an element at the beginning.



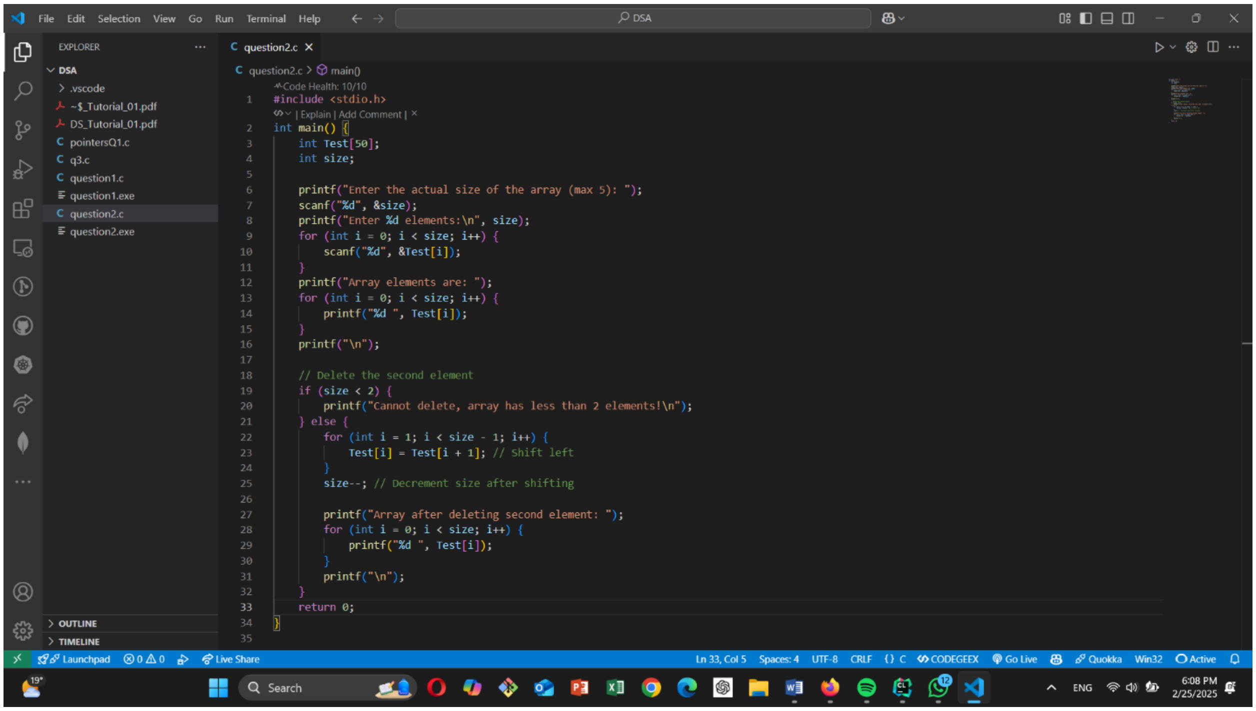
```c
#include <stdio.h>
int main() {
    int Test[50];
    int size, num, pos = 0; // pos = 0 (inserting at the beginning)

    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);
    if (size > 5 || size <= 0) {
        printf("Invalid size! Please enter a value between 1 and 5.\n");
        return 1;
    }

    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }

    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    printf("\n");

    printf("Enter the number to insert at the beginning: ");
    scanf("%d", &num);

    for (int i = size; i > pos; i--) {
        Test[i] = Test[i - 1];
    }
    Test[pos] = num;
    size++;
```

```
"D:\LENOVO\4 th sem\q3.exe"
Enter the actual size of the array (max 5):5

Enter 5 elements:
8

4

7

8

9

Array elements are: 8 4 7 8 9
Enter the number to insert at the beginning:2

Updated array elements are: 2 8 4 7 8 9
Process finished with exit code 0
```

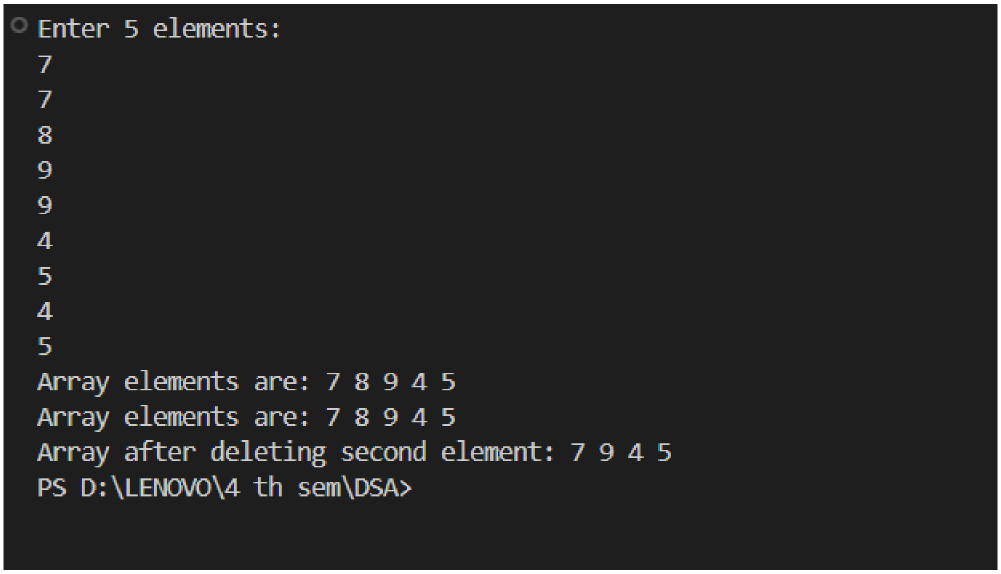


```
for (int i = size; i > pos; i--) {
    Test[i] = Test[i - 1];
}
Test[pos] = num;
size++;
printf( format: "Updated array elements are: ");
for (int i = 0; i < size; i++) {
    printf( format: "%d ", Test[i]);
}
return 0;
```

4) Write a program to delete the second element of array.

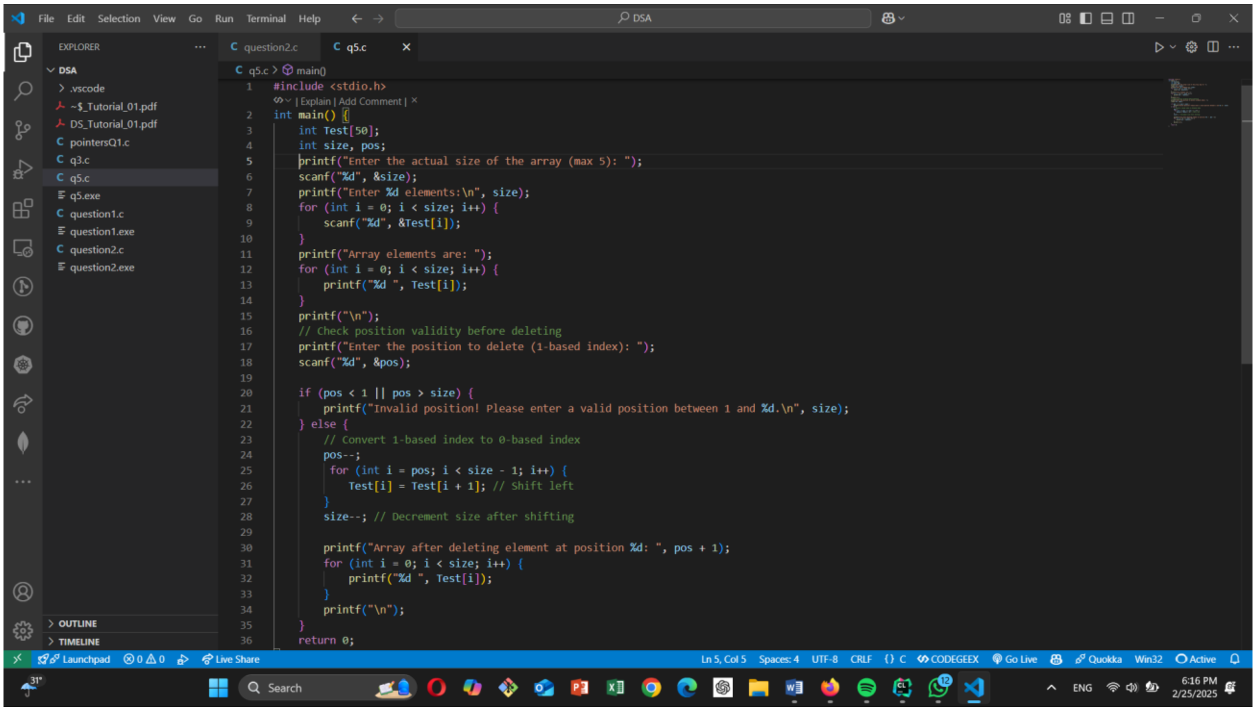


```
#include <stdio.h>
int main() {
    int Test[50];
    int size;

    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);
    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }
    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    printf("\n");

    // Delete the second element
    if (size < 2) {
        printf("Cannot delete, array has less than 2 elements!\n");
    } else {
        for (int i = 1; i < size - 1; i++) {
            Test[i] = Test[i + 1]; // Shift left
        }
        size--; // Decrement size after shifting

        printf("Array after deleting second element: ");
        for (int i = 0; i < size; i++) {
            printf("%d ", Test[i]);
        }
        printf("\n");
    }
    return 0;
}
```

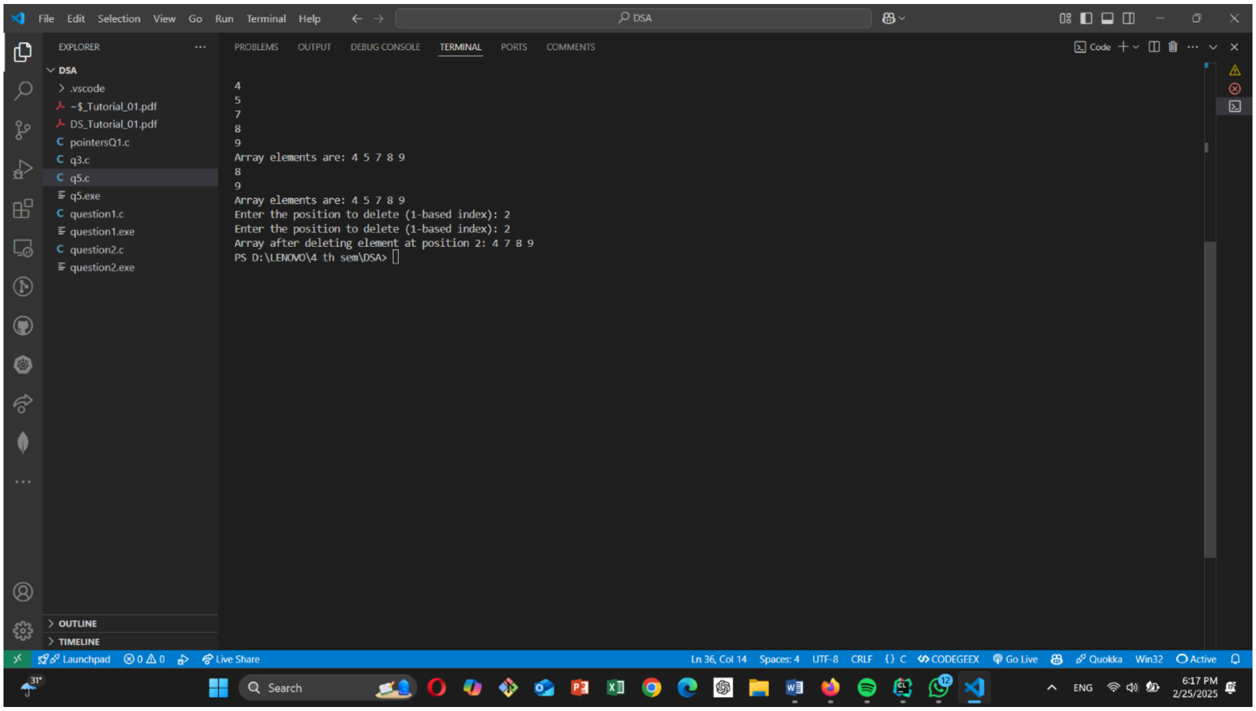


```
Enter 5 elements:
7
7
8
9
9
4
5
4
5
Array elements are: 7 8 9 4 5
Array elements are: 7 8 9 4 5
Array after deleting second element: 7 9 4 5
PS D:\LENOVO\4 th sem\DSA>
```

5)Write the code segment to check the position validity

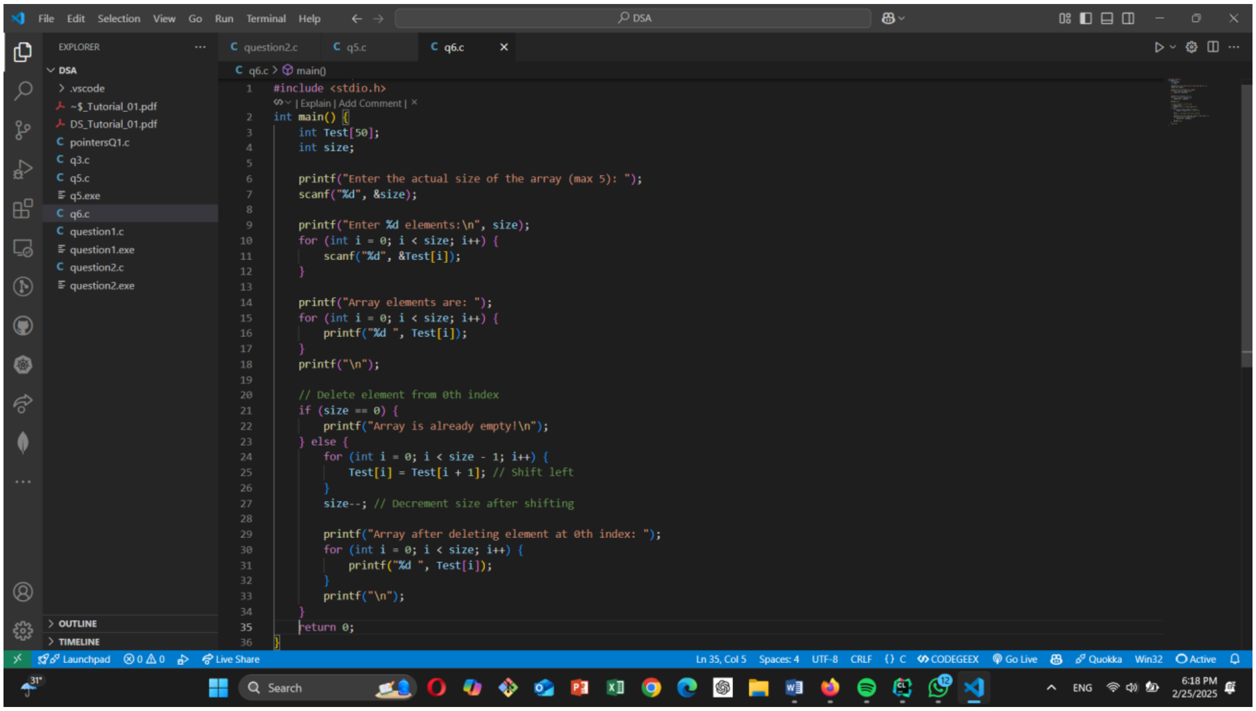


```c
#include <stdio.h>
int main() {
    int Test[50];
    int size, pos;
    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);
    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }
    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    printf("\n");
    // Check position validity before deleting
    printf("Enter the position to delete (1-based index): ");
    scanf("%d", &pos);

    if (pos < 1 || pos > size) {
        printf("Invalid position! Please enter a valid position between 1 and %d.\n", size);
    } else {
        // Convert 1-based index to 0-based index
        pos--;
         for (int i = pos; i < size - 1; i++) {
            Test[i] = Test[i + 1]; // Shift left
        }
        size--; // Decrement size after shifting

        printf("Array after deleting element at position %d: ", pos + 1);
        for (int i = 0; i < size; i++) {
            printf("%d ", Test[i]);
        }
        printf("\n");
    }
    return 0;
```

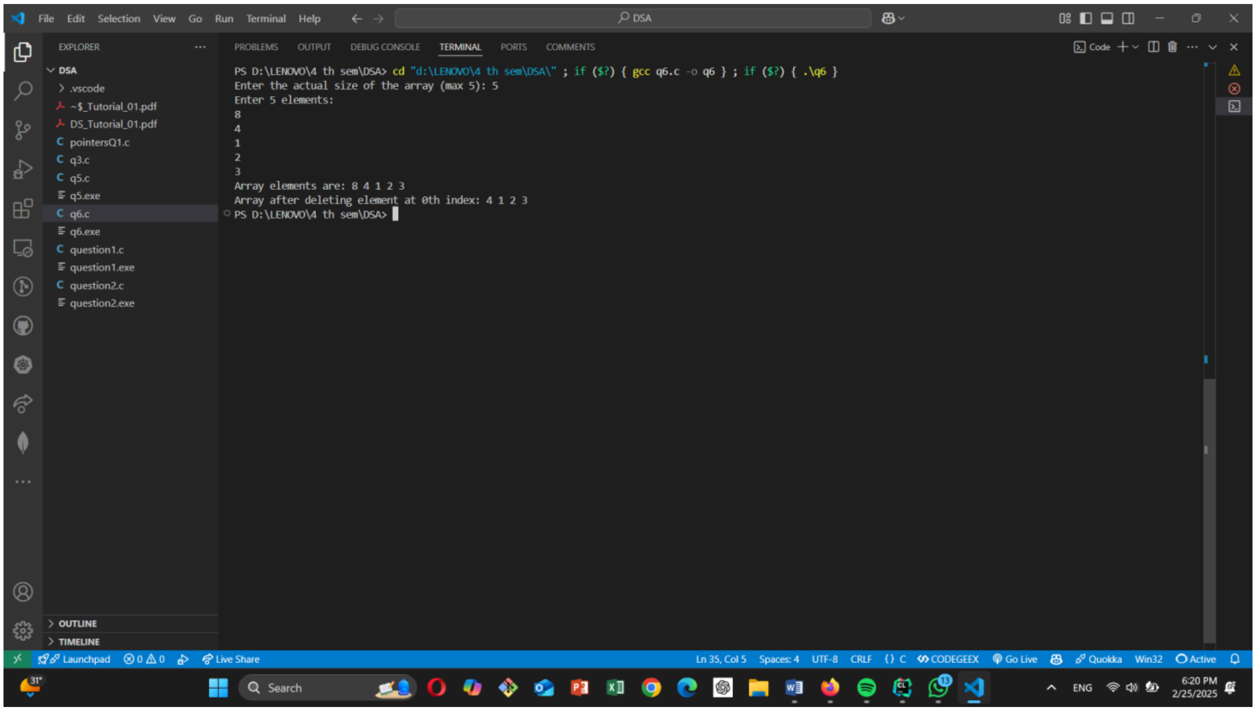


```
4
5
7
8
9
Array elements are: 4 5 7 8 9
8
9
Array elements are: 4 5 7 8 9
Enter the position to delete (1-based index): 2
Enter the position to delete (1-based index): 2
Array after deleting element at position 2: 4 7 8 9
PS D:\LENOVO\4 th sem\DSA> 
```

7). Modify the above program to delete the element from 0^{th} index.



```c
#include <stdio.h>
int main() {
    int Test[50];
    int size;

    printf("Enter the actual size of the array (max 5): ");
    scanf("%d", &size);

    printf("Enter %d elements:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &Test[i]);
    }

    printf("Array elements are: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", Test[i]);
    }
    printf("\n");

    // Delete element from 0th index
    if (size == 0) {
        printf("Array is already empty!\n");
    } else {
        for (int i = 0; i < size - 1; i++) {
            Test[i] = Test[i + 1]; // Shift left
        }
        size--; // Decrement size after shifting

        printf("Array after deleting element at 0th index: ");
        for (int i = 0; i < size; i++) {
            printf("%d ", Test[i]);
        }
        printf("\n");
    }
    return 0;
}
```



```
PS D:\LENOVO\4 th sem\DSA> cd "d:\LENOVO\4 th sem\DSA\" ; if ($?) { gcc q6.c -o q6 } ; if ($?) { .\q6 }
Enter the actual size of the array (max 5): 5
Enter 5 elements:
8
4
1
2
3
Array elements are: 8 4 1 2 3
Array after deleting element at 0th index: 4 1 2 3
PS D:\LENOVO\4 th sem\DSA>
```