

[Tutorial 01] - Index-225036C

One Dimensional Array

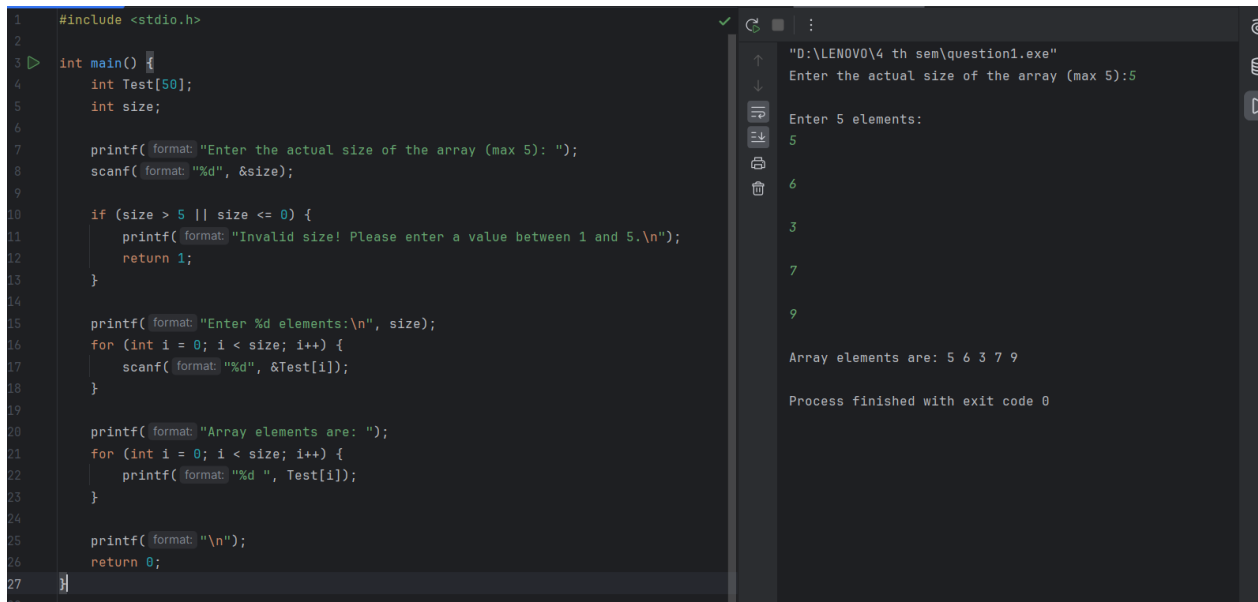
- 1). Write a C program to take inputs from user and print the array values using for loops. (For one-D Array)

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**



```
1 #include <stdio.h>
2
3 int main() {
4     int Test[50];
5     int size;
6
7     printf( format: "Enter the actual size of the array (max 5): ");
8     scanf( format: "%d", &size);
9
10    if (size > 5 || size <= 0) {
11        printf( format: "Invalid size! Please enter a value between 1 and 5.\n");
12        return 1;
13    }
14
15    printf( format: "Enter %d elements:\n", size);
16    for (int i = 0; i < size; i++) {
17        scanf( format: "%d", &Test[i]);
18    }
19
20    printf( format: "Array elements are: ");
21    for (int i = 0; i < size; i++) {
22        printf( format: "%d ", Test[i]);
23    }
24
25    printf( format: "\n");
26    return 0;
27 }
```

Output:

```
"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.

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

"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.

```
1  #include <stdio.h>
2  int main() {
3      int Test[50];
4      int size, num, pos = 0; // pos = 0 (inserting at the beginning)
5
6      printf( format: "Enter the actual size of the array (max 5): ");
7      scanf( format: "%d", &size);
8      if (size > 5 || size <= 0) {
9          printf( format: "Invalid size! Please enter a value between 1 and 5.\n");
10         return 1;
11     }
12
13     printf( format: "Enter %d elements:\n", size);
14     for (int i = 0; i < size; i++) {
15         scanf( format: "%d", &Test[i]);
16     }
17
18     printf( format: "Array elements are: ");
19     for (int i = 0; i < size; i++) {
20         printf( format: "%d ", Test[i]);
21     }
22     printf( format: "\n");
23
24     printf( format: "Enter the number to insert at the beginning: ");
25     scanf( format: "%d", &num);
26
27     for (int i = size; i > pos; i--) {
28         Test[i] = Test[i - 1];
29     }
30     Test[pos] = num;
31     size++;
32
33     printf( format: "Updated array elements are: ");
34     for (int i = 0; i < size; i++) {
35         printf( format: "%d ", Test[i]);
36     }
37     return 0;}

"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.

The screenshot shows a Visual Studio Code editor window with a C program titled 'question2.c'. The program is designed to delete the second element of an array. It starts by including `<stdio.h>` and defining the `main` function. An array `Test` of size 50 is declared, and `size` is initialized. The program prompts the user to enter the actual size of the array (maximum 5). It then reads the size and the array elements using `scanf`. After displaying the array elements, it checks if the size is less than 2. If not, it shifts all elements from index 1 onwards one position to the left and decrements the size. Finally, it prints the array after deletion and returns 0.

```

1  #include <stdio.h>
2  int main()
3  {
4      int Test[50];
5      int size;
6
7      printf("Enter the actual size of the array (max 5): ");
8      scanf("%d", &size);
9      printf("Enter %d elements:\n", size);
10     for (int i = 0; i < size; i++) {
11         scanf("%d", &Test[i]);
12     }
13     printf("Array elements are: ");
14     for (int i = 0; i < size; i++) {
15         printf("%d ", Test[i]);
16     }
17     printf("\n");
18
19     // Delete the second element
20     if (size < 2) {
21         printf("Cannot delete, array has less than 2 elements!\n");
22     } else {
23         for (int i = 1; i < size - 1; i++) {
24             Test[i] = Test[i + 1]; // shift left
25         }
26         size--; // Decrement size after shifting
27
28         printf("Array after deleting second element: ");
29         for (int i = 0; i < size; i++) {
30             printf("%d ", Test[i]);
31         }
32         printf("\n");
33     }
34     return 0;
35 }

```

```

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

```

File Edit Selection View Go Run Terminal Help
DSA
EXPLORER
  DSA
    .vscode
    ~$.Tutorial_01.pdf
    DS_Tutorial_01.pdf
    pointersQ1.c
    q3.c
    q5.c
    q5.exe
    question1.c
    question1.exe
    question2.c
    question2.exe
  OUTLINE
  TIMELINE
C question2.c  C q5.c
C q5.c main()
1 #include <stdio.h>
2 int main() {
3     int Test[50];
4     int size, pos;
5     printf("Enter the actual size of the array (max 5): ");
6     scanf("%d", &size);
7     printf("Enter %d elements:\n", size);
8     for (int i = 0; i < size; i++) {
9         scanf("%d", &Test[i]);
10    }
11    printf("Array elements are: ");
12    for (int i = 0; i < size; i++) {
13        printf("%d ", Test[i]);
14    }
15    printf("\n");
16    // Check position validity before deleting
17    printf("Enter the position to delete (1-based index): ");
18    scanf("%d", &pos);
19
20    if (pos < 1 || pos > size) {
21        printf("Invalid position! Please enter a valid position between 1 and %d.\n", size);
22    } else {
23        // Convert 1-based index to 0-based index
24        pos--;
25        for (int i = pos; i < size - 1; i++) {
26            Test[i] = Test[i + 1]; // Shift left
27        }
28        size--; // Decrement size after shifting
29
30        printf("Array after deleting element at position %d: ", pos + 1);
31        for (int i = 0; i < size; i++) {
32            printf("%d ", Test[i]);
33        }
34        printf("\n");
35    }
36    return 0;

```

The screenshot shows a Visual Studio Code editor with a terminal window open. The terminal displays the output of a C program that prints an array of numbers (4 5 7 8 9), prompts the user to enter a position to delete (1-based index), and then prints the array after deletion. The user has entered '2', and the output shows the array after deleting the element at position 2: 4 7 8 9.

```
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 0th index.

```
1 #include <stdio.h>
2 int main() {
3     int Test[50];
4     int size;
5
6     printf("Enter the actual size of the array (max 5): ");
7     scanf("%d", &size);
8
9     printf("Enter %d elements:\n", size);
10    for (int i = 0; i < size; i++) {
11        scanf("%d", &Test[i]);
12    }
13
14    printf("Array elements are: ");
15    for (int i = 0; i < size; i++) {
16        printf("%d ", Test[i]);
17    }
18    printf("\n");
19
20    // Delete element from 0th index
21    if (size == 0) {
22        printf("Array is already empty!\n");
23    } else {
24        for (int i = 0; i < size - 1; i++) {
25            Test[i] = Test[i + 1]; // Shift left
26        }
27        size--; // Decrement size after shifting
28
29        printf("Array after deleting element at 0th index: ");
30        for (int i = 0; i < size; i++) {
31            printf("%d ", Test[i]);
32        }
33        printf("\n");
34    }
35    return 0;
36 }
```

```
PS D:\LEHNOVO\4 th sem\DSA> cd "d:\LEHNOVO\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:\LEHNOVO\4 th sem\DSA>
```