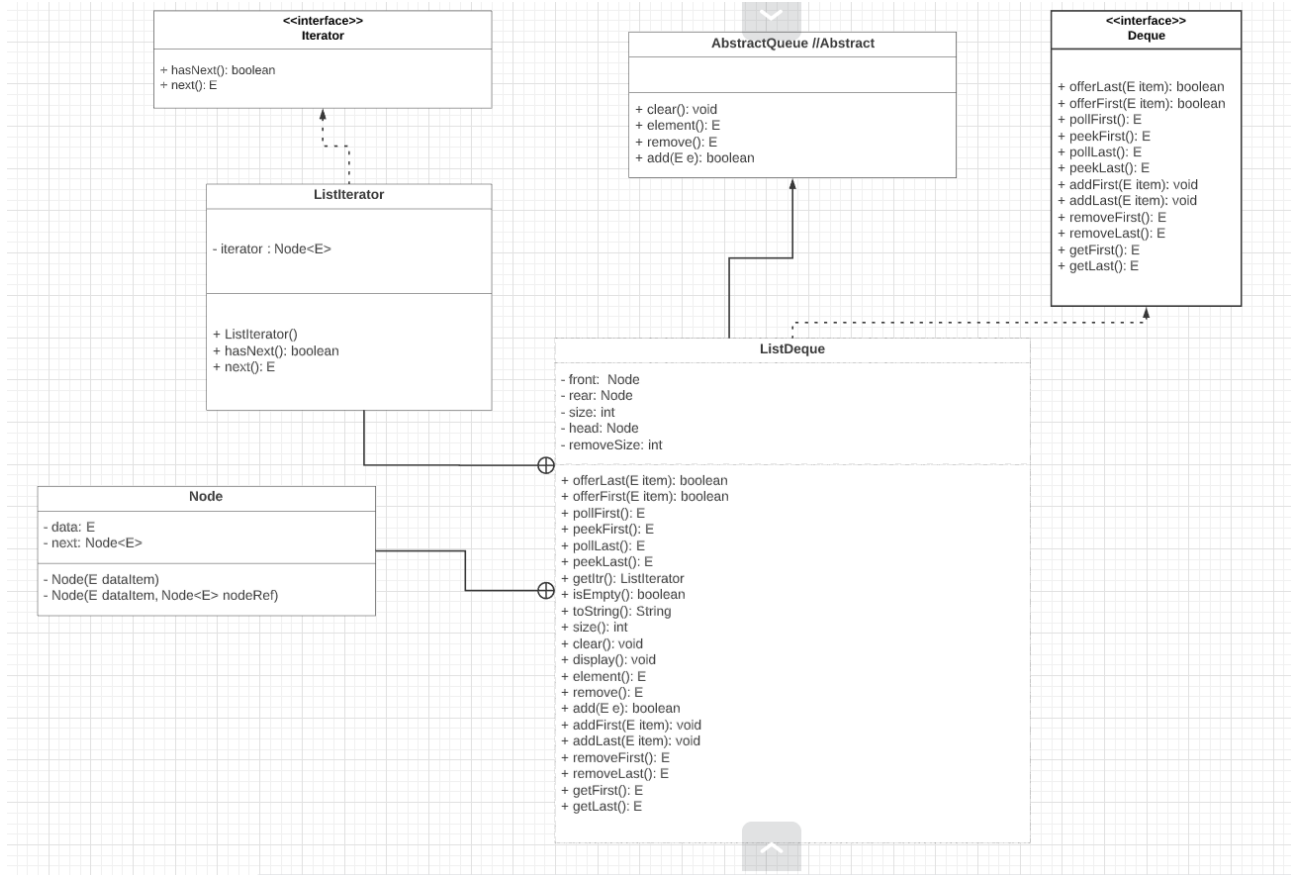


**GTU Department of Computer Engineering**  
**CSE 222/505 - Spring 2020**  
**Homework 4 Report**

Mert Can BEŞİRLİ  
1801042663

## #Question 2#

### 1. CLASS DIAGRAMS



### 2. PROBLEM SOLUTION APPROACH

Implement a `ListDeque` class which `Deque` interface, extends `AbstractQueue` class. Our problem here is to add and remove elements from the beginning and the end in `Deque` (`Deque` is `Queue`). We have two linked lists in the `ListDeque` class. One for the elements in deque and the other to keep the nodes removed. You should use a removed node, if any available, when a new node is needed instead of creating a new node. Here we have added and subtracted from the beginning and the end with the `offerFirst()`, `offerLast()`, `pollFirst()`, `pollLast()` method to it in the `ListDeque` class. `Front` and `rear` keeping the `Deque` elements. `Head` keeping removing node. We hold the nodes that are deleted `pollFirst()` and `pollLast()` with the `head`. If necessary, we do not create a new node again. We made our test in integer type but it can work in other types because its generic.

### 3. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Excepted Results	Actual Results	Pass/Fail
T1	Check Offer First to Deque valid data	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	A Enter element for add first deque 1	Deque of elements: 1	As Expected	Pass
T2	Check Offer First to Deque to invalid data	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	A Enter element for add first deque sadsd	java.util.Inp utMismatch Exception	As Expected	Fail
T3	Check Offer Last to Deque to valid data	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First	B Enter element for add last deque 3	B Enter element for add last deque 3	As Expected	Pass

		to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit				
T4	Check Offer Last to Deque to invalid data	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	Check Offer Last to Deque to valid data	java.util.Inp utMismatch Exception	As Expected	Fail
T5	Check Deque Size in the beginning	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	E	Size: 0	As Expected	Pass
T6	Deque is 1 2 3 4 then	Choice: A->Offer	C	Remove element: 1	As Expected	Pass

	Poll First	First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit		Deque of elements: 2 3 4		
T7	Deque is 1 2 3 4 then Poll Last	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	D	Remove element: 4 Deque of elements: 1 2 3	As Expected	Pass
T8	Deque is 1 2 3 4 then 2x Poll Last then Display Deque Size	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and	E	Size: 2	As Expected	Pass

		Removed Node Display Q->Quit				
T9	Deque is 1 2 3 4 then 2x Poll First then Display Deque and Removing List	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	F	Deque of elements: 3 4  Removing list nodes  Head: 1 Head: 2	As Expected	Pass
T10	Deque is 1 2 then 2x Poll Last then Offer Last 3	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	B Enter element for add last deque 1 Deque of elements: 1  B Enter element for add last deque 2 Deque of elements: 1 2  D Remove element: 2 Deque of elements: 1	Deque of elements: 3	As Expected	Pass

			D Remove element: 2 Deque of elements: 1  B Enter element for add last deque 3			
T11	Check Quit	Choice: A->Offer First to Deque B->Offer Last to Deque C->Poll First to Deque D->Poll Last to Deque E->Deque size F->Deque and Removed Node Display Q->Quit	Q		As Expected	Pass

## 4. RUNNING AND RESULTS

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Queue$ java Test
```

Choice:

A->Offer First to Deque

B->Offer Last to Deque

C->Poll First to Deque

D->Poll Last to Deque

E->Deque size

F->Deque and Removed Node Display

Q->Quit

A

Enter element for add first deque

1

Deque of elements:

1

Choice:

A->Offer First to Deque

B->Offer Last to Deque

C->Poll First to Deque

D->Poll Last to Deque

E->Deque size

F->Deque and Removed Node Display

Q->Quit

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Queue$ java Test
```

Choice:

A->Offer First to Deque

B->Offer Last to Deque

C->Poll First to Deque

D->Poll Last to Deque

E->Deque size

F->Deque and Removed Node Display

Q->Quit

A

Enter element for add first deque

sadsd

java.util.InputMismatchException

Choice:

A->Offer First to Deque

B->Offer Last to Deque

C->Poll First to Deque

D->Poll Last to Deque

E->Deque size

F->Deque and Removed Node Display

Q->Quit



```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Queue$ java Test
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
B
Enter element for add last deque
3
Deque of elements:
3

Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
B
Enter element for add last deque
1,2
java.util.InputMismatchException

Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Queue$ java Test
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
E
Size: 0

Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
A
Enter element for add first deque
4
Deque of elements:
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
A
Enter element for add first deque
3
Deque of elements:
3
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
A
Enter element for add first deque
2
Deque of elements:
2
3
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
A
Enter element for add first deque
1
Deque of elements:
1
2
3
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
C
Remove element: 1
Deque of elements:
2
3
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
D
Remove element: 4
Deque of elements:
1
2
3
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
D
Remove element: 3
Deque of elements:
1
2
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
D
Remove element: 3
Deque of elements:
1
2
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
E
Size: 2
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
C
Remove element: 2
Deque of elements:
3
4
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
F
Deque of elements:
3
4
```

Removing list nodes

```
Head: 1
Head: 2
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
B
Enter element for add last deque
1
Deque of elements:
1
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
B
Enter element for add last deque
2
Deque of elements:
1
2
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
D
Remove element: 2
Deque of elements:
1
```

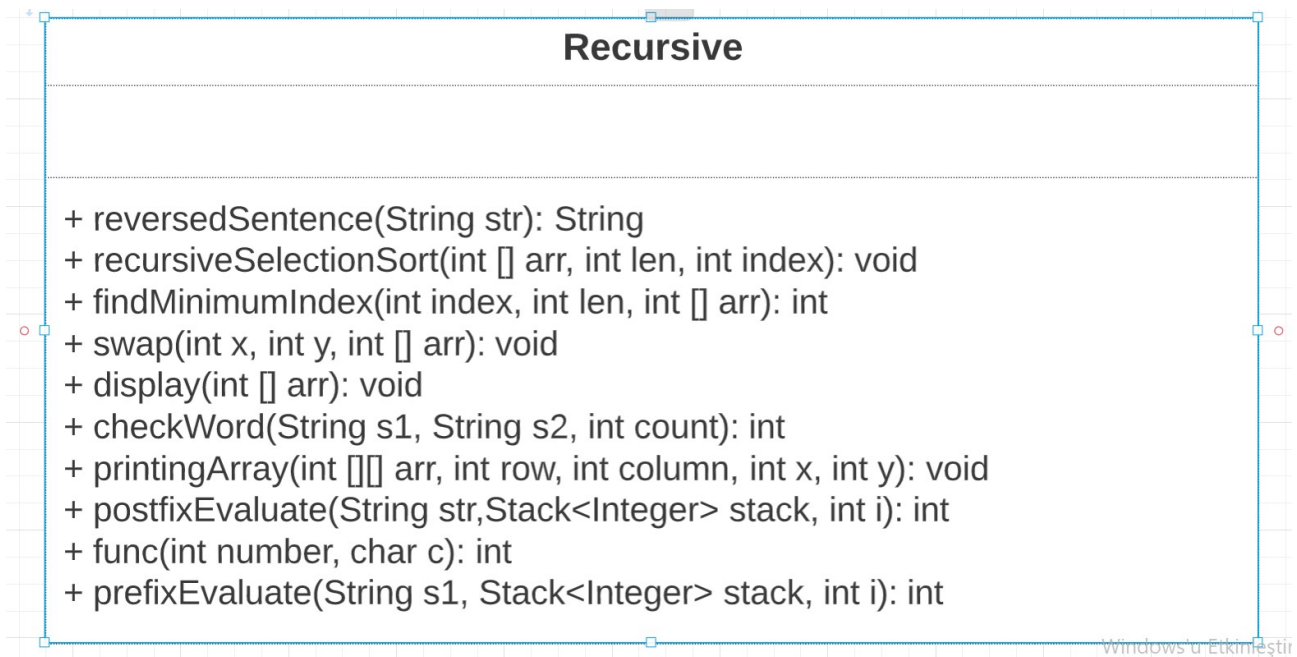
```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
D
Remove element: 1
Deque of elements:
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
B
Enter element for add last deque
3
Deque of elements:
3
```

```
Choice:
A->Offer First to Deque
B->Offer Last to Deque
C->Poll First to Deque
D->Poll Last to Deque
E->Deque size
F->Deque and Removed Node Display
Q->Quit
Q
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Queue$
```

## #Question 3#

### 1. CLASS DIAGRAMS



### 2. PROBLEM SOLUTION APPROACH

In our first problem, we get a sentence or word from the user. Then the method is called and this String statement we get from the user goes together. When the string expression is null or empty as base case, the string proceeds by breaking the string.

In our second problem, we check whether the word the user enters is elfish. First, the user must enter a word and then the method must be called. The method contains the word the user enters, a string variable with the elf and count. We check the letter letter by letter. Continue until the last letter Count is incremented whenever it comes to the expression 'e', 'l', 'f'. When the function is over, if count is equal to the length of the string variable where the elf is, the word becomes elfish.

In our third problem, we sort the array elements according to the selection algorithm. I determined the length of the series as seven. First, we need to get seven elements from the user. Then the method is called. In the method, an array with an array with a length and zero goes. As base case, the method stops if the index is equal to the length of the array. First, the index with the minimum number in the array is found. Then the index is replaced with this minimum index that corresponds to this array. The index increases by one. After the method is finished, the method to print the array elements is called.

In our fourth problem, the evaluation process of the prefix expression. I gave 2 examples for the prefix expression, we find the result in both. In the method, the

length of the prefix expression, stack, and prefix expression goes short. We go from the end of the prefix statement to the beginning. First, the last character of the String expression is taken. Check whether this is a space, a number or \*, -, +, /.

If the character is blank, the length is reduced by one. If the number is pushed to stack. If the expression is, the numbers in the stack are processed according to the expression. The length of the string character as base case is -1.

In our fifth problem, the evaluation process of the postfix expression. I gave 2 examples for the postfix expression, we find the result in both. In the method, postfix expression, stack and i = 0 goes. The string starts at the beginning of the statement, and i represents the beginning. Each process is progressed character by character. The character is taken first. The character, space, number, and expression can be \*, -, /, +. If space is, i will increase by one. The number is pushed to stack, but when there is a two-digit number, it is processed accordingly. If the expression is the stack pops up and the expression is applied to the numbers. And if i increases one, the function ends when the string is equal to the length of the expression. This is the base case.

In the sixth problem, we are asked to print the array elements as on the screen.

First, the number of rows and columns are taken from the user and the method is called. In the method, the variables x = 0 and y = 0 also go. If x or y exceeds the row and column, the function ends. This base case.

In other cases, first line, first column and other elements are printed according to row and column conditions. Last row and column 1 decreases, x and y increase 1.

I wrote them in our code as a menu.

### 3.TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Excepted Results	Actual Results	Pass/Fail
T1	Enter choice First Problems	Enter of choice between 1-6 problems, 7 is exit:	1 Enter the sentence for reverse it: writer	Reversed the sentence is: retirw	As Expected	Pass
T2	Enter choice Second Problem valid data	Enter of choice between 1-6 problems, 7 is exit:	2 Enter the word for elfish or not:	whiteleaf This word is elfish: whiteleaf	As Expected	Pass
T3	Enter choice Second Problem invalid data	Enter of choice between 1-6 problems, 7	2 Enter the word for elfish or not:	writer This word is not elfish	As Expected	Fail



		is exit:				
T4	Enter choice Third Problem valid data	Enter of choice between 1-6 problems, 7 is exit:	3 Enter of array elements for selecting sorting:	87 5 6 1 2 0 9 0 1 2 5 6 9 87	As Expected	Pass
T5	Enter choice Third Problem invalid data	Enter of choice between 1-6 problems, 7 is exit:	3 Enter of array elements for selecting sorting:	1 2 r java.util.Inp utMismatch Exception 0 0 0 0 0 1 2	As Expected	Fail
T6	Enter choice Fourth Problem	Enter of choice between 1-6 problems, 7 is exit:	4	Evaluating prefix is : -3 Evaluating prefix is : 2	As Expected	Pass
T7	Enter choice Fifth Problem	Enter of choice between 1-6 problems, 7 is exit:	5	Evaluating postfix is : 32 Evaluating postfix is : 62	As Expected	Pass
T8	Enter choice Sixth Problem	Enter of choice between 1-6 problems, 7 is exit:	6 Enter array row and column:	3 2 1 2 4 6 5 3	As Expected	Pass
T9	Quit choice	Enter of choice between 1-6 problems, 7 is exit:	7	Exit...	As Expected	Pass

## 4. RUNNING AND RESULTS



```
ubuntu@ubuntu-VirtualBox:~$ cd Masaüstü/List/Recursive/
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
1
Enter the sentence for reverse it:
writer
Reversed the sentence is: retirw
Enter of choice between 1-6 problems, 7 is exit:
```

```
Enter of choice between 1-6 problems, 7 is exit:
2
Enter the word for elfish or not:
whiteleaf
This word is elfish: whiteleaf
Enter of choice between 1-6 problems, 7 is exit:
2
Enter the word for elfish or not:
writer
This word is not elfish
Enter of choice between 1-6 problems, 7 is exit:
█
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
3
Enter of array elements for selecting sorting:
87
5
6
1
2
0
9
0 1 2 5 6 9 87
Enter of choice between 1-6 problems, 7 is exit:
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
3
Enter of array elements for selecting sorting:
1
2
r
java.util.InputMismatchException
0 0 0 0 0 1 2
Enter of choice between 1-6 problems, 7 is exit:
█
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
4
Evaluating prefix is : -3
Evaluating prefix is : 2
Enter of choice between 1-6 problems, 7 is exit:

```

Ubuntu Yazılımlar

```
^Cubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
5
Evaluating postfix is : 32
Evaluating postfix is : 62
Enter of choice between 1-6 problems, 7 is exit:

```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$ java Test
Enter of choice between 1-6 problems, 7 is exit:
6
Enter array row and column:
3
2
1 2 4 6 5 3
Enter of choice between 1-6 problems, 7 is exit:
6
Enter array row and column:
4
4
1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
Enter of choice between 1-6 problems, 7 is exit:
6
Enter array row and column:
2
2
1 2 4 3
Enter of choice between 1-6 problems, 7 is exit:
6
Enter array row and column:
5
4
1 2 3 4 8 12 16 20 19 18 17 13 9 5 6 7 11 15 14 10
Enter of choice between 1-6 problems, 7 is exit:
```

```
Enter of choice between 1-6 problems, 7 is exit:
7
Exit...
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/Recursive$
```

