

--- ****Output of 1st program****

-
1. Create an array
 2. Display an array
 3. Insert an element into an array
 4. Delete an element
 5. Exit
-

Enter your choice: 1
Enter the size of the array: 5
Enter the array elements: 1 2 3 4 5

1. Create an array
 2. Display an array
 3. Insert an element into an array
 4. Delete an element
 5. Exit
-

Enter your choice: 2
Array elements are: 1 2 3 4 5

1. Create an array
 2. Display an array
 3. Insert an element into an array
 4. Delete an element
 5. Exit
-

Enter your choice: 3
Enter the number to be inserted: 6
Enter the position to insert the element (1-based index): 4

1. Create an array
 2. Display an array
 3. Insert an element into an array
 4. Delete an element
 5. Exit
-

Enter your choice: 2
Array elements are: 1 2 3 6 4 5

1. Create an array
 2. Display an array
 3. Insert an element into an array
 4. Delete an element
 5. Exit
-

Enter your choice: 4

Enter the position of the element to delete (1-based index): 4

Deleted element is: 6

1. Create an array
2. Display an array
3. Insert an element into an array
4. Delete an element
5. Exit

Enter your choice: 2

Array elements are: 1 2 3 4 5

1. Create an array
2. Display an array
3. Insert an element into an array
4. Delete an element
5. Exit

Enter your choice: 3

Enter the number to be inserted: 7

Enter the position to insert the element (1-based index): 7

Insertion is not allowed.

1. Create an array
2. Display an array
3. Insert an element into an array
4. Delete an element
5. Exit

Enter your choice: 4

Enter the position of the element to delete (1-based index): 6

Invalid Position.

1. Create an array
2. Display an array
3. Insert an element into an array
4. Delete an element
5. Exit

Enter your choice: 5

End of program.

--- **Output of 2nd program**

Enter main string: Hello everyone
Enter pattern string: everyone
Enter replacing string: world
The resultant string is: Hello world

Enter main string: Hello everyone
Enter pattern string: Everyone
Enter replacing string: world
Pattern not found.

--- **Output of 3rd program**

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 1

Enter the element to push: 1

Element 1 pushed onto the stack.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 1

Enter the element to push: 2

Element 2 pushed onto the stack.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 1

Enter the element to push: 3
Element 3 pushed onto the stack.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 5

Current stack elements: 1 2 3

Top of the stack is at position 2 with value 3.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 2

Popped element: 3

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 5

Current stack elements: 1 2

Top of the stack is at position 1 with value 2.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 3

The stack is not a palindrome.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack

3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 4

Stack is not full. No overflow.

Stack is not empty. No underflow.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 2

Popped element: 2

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 5

Current stack elements: 1

Top of the stack is at position 0 with value 1.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 3

The stack is a palindrome.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 2

Popped element: 1

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 4

Stack is not full. No overflow.

Stack is empty! Underflow condition exists.

Menu:

1. Push an element onto the stack
2. Pop an element from the stack
3. Check if the stack is a palindrome
4. Demonstrate overflow / underflow situations
5. Display the status of the stack
6. Exit

Enter your choice: 6

End of Program.

--- **Output of 4th program**

Enter a valid infix epression: a+b

Infix expression is: a+b

Postfix expression is: ab+

Enter a valid infix epression: (A+B*C)/(D-E)*(F*G)+(H/I/J)

Infix expression is: (A+B*C)/(D-E)*(F*G)+(H/I/J)

Postfix expression is: ABC*+DE-/FG**HI/J/+

--- ****Output of 5th program****

Application of stacks

1. Evaluation of postfix expression
2. Tower of hanoi
3. Exit

Enter your choice: 1

Enter a valid postfix Expression: 12+

The result of postfix expression 12+ is = 3

Application of stacks

1. Evaluation of postfix expression
2. Tower of hanoi
3. Exit

Enter your choice: 1

Enter a valid postfix Expression: 99*99*+99*+99*+

The result of postfix expression 99*99*+99*+99*+ is = 324

Application of stacks

1. Evaluation of postfix expression
2. Tower of hanoi
3. Exit

Enter your choice: 2

Enter number of disks: 3

Move disk 1 from S to D

Move disk 2 from S to A

Move disk 1 from D to A

Move disk 3 from S to D

Move disk 1 from A to S

Move disk 2 from A to D

Move disk 1 from S to D

Number of moves is 7

Application of stacks

1. Evaluation of postfix expression
2. Tower of hanoi
3. Exit

Enter your choice: 3

End of Program.

--- **Output of 6th program**

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 1

Enter element to be inserted: 1

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 1

Enter element to be inserted: 2

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 1

Enter element to be inserted: 3

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 3

Elements of the queue are: 1 2 3

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 2

Deleted element is 1

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 3

Elements of the queue are: 2 3

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 2

Deleted element is 2

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 2

Deleted element is 3

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 2

Queueu is Empty

Circular Queue Operations

- 1.Enqueue
- 2.Dequeue
- 3.Display
- 4.Exit

Enter your choice: 4

End of Program.