G. H. Raisoni College Of Engineering And Management, Wagholi Pune 2021- 2022 Assignment no :- 5			
Department	CE [SUMMER 2022 (Online)]		
Term / Section	III/B	Date Of **submission**	20-10-2021
Subject Name /Code	Data Structures and Algorithms/ UCSL201/UCSP201		
Roll No.	SCOB77	Name	Pratham Rajkumar pitty
Registration Number	2020AC0E1100107		

Experiment No.5

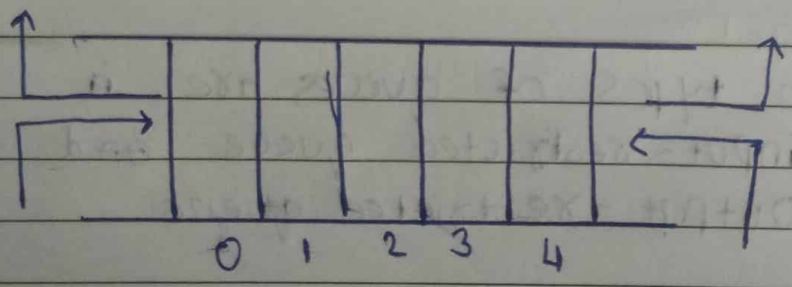
Aim: A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate ~~representation mapping a deque~~ deque with Functions to add and delete elements from either end of the deque.

Objective :- To Simulate deque with Functions to add and delete elements from either end of the deque.

Theory :

Double-Ended queue (Deque)

In this queue The insertion takes place from one end while the deletion takes place from another end. The end at which the insertion occurs is known as the rear-end whereas the end at which the deletion occurs is known as "Front end".

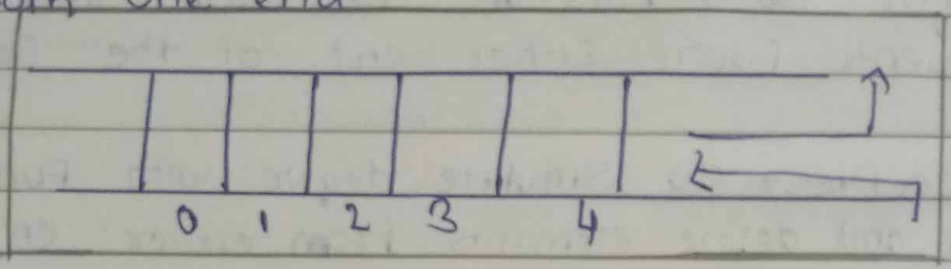


Deque is a linear data structure in which the insertion and deletion operations are performed from both ends. We can say that deque is a generalized version of the queue.

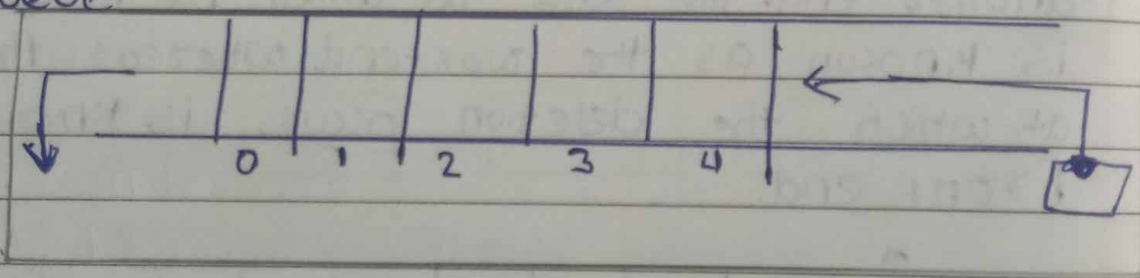
- Deque can be used both as stack and queue as it allows the insertion and deletion operation.

on both ends.

deque can be considered as a stack because In deque, the insertion and deletion operation can be performed from one side. The stack follows the LIFO rule in which both the insertion and deletion can be performed only from one end

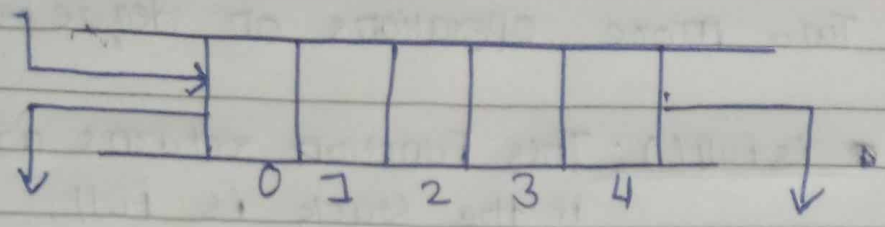


In deque, the insertion can be performed on one end, and the deletion can be done on another end. The queue follows the FIFO rule in which the element is inserted on one end and deleted from another end. Therefore we conclude that the deque can also be considered as the queue



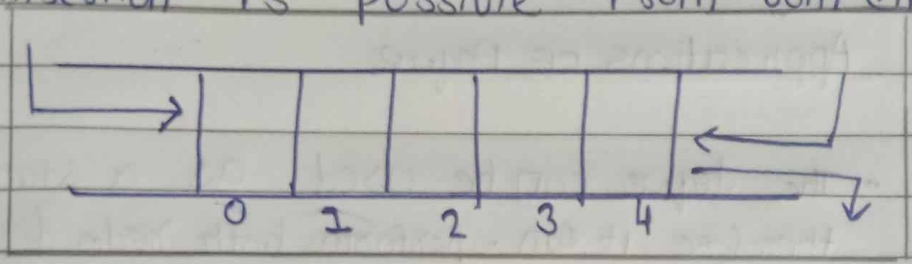
Two types of queues are,
(1). input - restricted queue
(2). output - restricted queue

1. input - restricted queue : The input - restricted queue means that some restrictions are applied to the insertion. In input - resisted queue the insertion is applied to one end while the deletion is applied from both the end



2. Output -restricted queue :

The Output -restricted queue means that some restrictions are applied to the deletion operation. In an output -restricted queue, the deletion can be applied only from one end, whereas the insertion is possible from both ends



Operations on Deque

The following are the operations applied on deque:

- Insert at front
- Insert at rear
- Delete at front
- Delete at rear

Peak operation in deque

- Using peak operation we can get the front and the rear element of the dequeue.

Two more operations on deque are

- isfull(): This Function returns a true value if the stack is full; otherwise, it returns a false value.

- isEmpty(): This Function returns a true value if the Stack is empty; otherwise, it returns a false value.

Applications of Deque

- The deque can be used as a stack and q therefore, it can perform both redo & undo operat

- It can be used as a palindrome checker means that if we read the string from both ends, then the string would be the same

- It can be used as For multi-processor Scheduling

Program code

// AIM :- A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end.

// Obtain a data representation mapping a deque into a one-dimensional array.

// Write C++ program to simulate deque with functions to add and delete elements from either end of the deque.

```
#include <iostream>

using namespace std;

#define SIZE 5


class dequeue

{


   int a[10], front, rear, count;


public:

   dequeue();

   void add_at_beg(int);

   void add_at_end(int);

   void delete_fr_front();

   void delete_fr_rear();

   void display();

};

dequeue::dequeue()

{

   front = -1;

   rear = -1;

   count = 0;

}
```

```
void dequeue::add_at_beg(int item)
{
   int i;
   if (front == -1)
   {
      front++;
      rear++;
      a[rear] = item;
      count++;
   }
   else if (rear >= SIZE - 1)
   {
      cout << "\nInsertion is not possible,overflow!!!!";
   }
   else
   {
      for (i = count; i >= 0; i--)
      {
         a[i] = a[i - 1];
      }
      a[i] = item;
      count++;
      rear++;
   }
}
void dequeue::add_at_end(int item)
{

   if (front == -1)
   {
      front++;
```

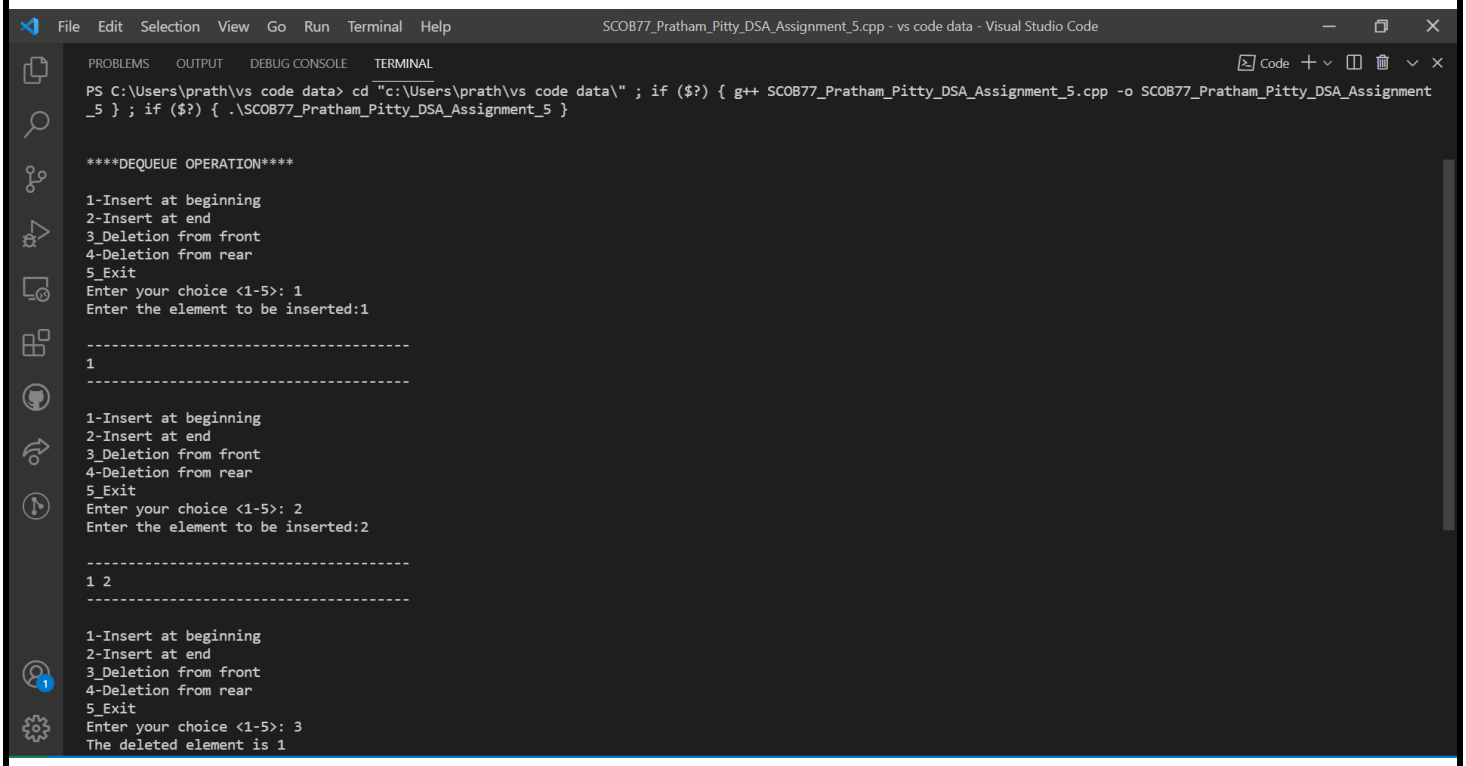
```
        rear++;
        a[rear] = item;
        count++;
    }
    else if (rear >= SIZE - 1)
    {
        cout << "\nInsertion is not possible,overflow!!!";
        return;
    }
    else
    {
        a[++rear] = item;
    }
}

void dequeue::display()
{
    for (int i = front; i <= rear; i++)
    {
        cout << a[i] << " ";
    }
}

void dequeue::delete_fr_front()
{
    if (front == -1)
    {
        cout << "Deletion is not possible:: Dequeue is empty";
        return;
    }
    else
```

```
    {
        if (front == rear)
        {
            front = rear =  -1;
            return;
        }
        cout << "The deleted element is " << a[front];
        front = front + 1;
    }
}

void dequeue::delete_fr_rear()
{
    if (front == -1)
    {
        cout << "Deletion is not possible: Dequeue is empty";
        return;
    }
    else
    {
        if (front == rear)
        {
        cout << "The deleted element is " << a[rear];
            front = rear =-1;
        }
        rear = rear -1;
    }
    // display();
}

int main()
```

```
{
    int c, item;
    dequeue d1;

    cout << "\n\n****DEQUEUE OPERATION****\n";
    do
    {
        cout << "\n1-Insert at beginning";
        cout << "\n2-Insert at end";
        cout << "\n3_Deletion from front";
        cout << "\n4-Deletion from rear";
        cout << "\n5_Exit";
        cout << "\nEnter your choice <1-5>: ";
        cin >> c;

        switch (c)
        {
        case 1:
            cout << "Enter the element to be inserted:";
            cin >> item;
            d1.add_at_beg(item);
            cout << "\n---------------------------------------\n";
            d1.display();
            cout << "\n---------------------------------------\n";
            break;

        case 2:
            cout << "Enter the element to be inserted:";
            cin >> item;
            d1.add_at_end(item);
            cout << "\n---------------------------------------\n";
```

```
            d1.display();
            cout << "\n---------------------------------------\n";
            break;

        case 3:
            d1.delete_fr_front();
            cout << "\n---------------------------------------\n";
            d1.display();
            cout << "\n---------------------------------------\n";
            break;
        case 4:
            d1.delete_fr_rear();
            cout << "\n---------------------------------------\n";
            d1.display();
            cout << "\n---------------------------------------\n";
            break;

        case 5:
            exit(1);
            break;

        default:
            cout << "Invalid choice";
            break;
        }

    } while (c != 6);
    return 0;
}
```

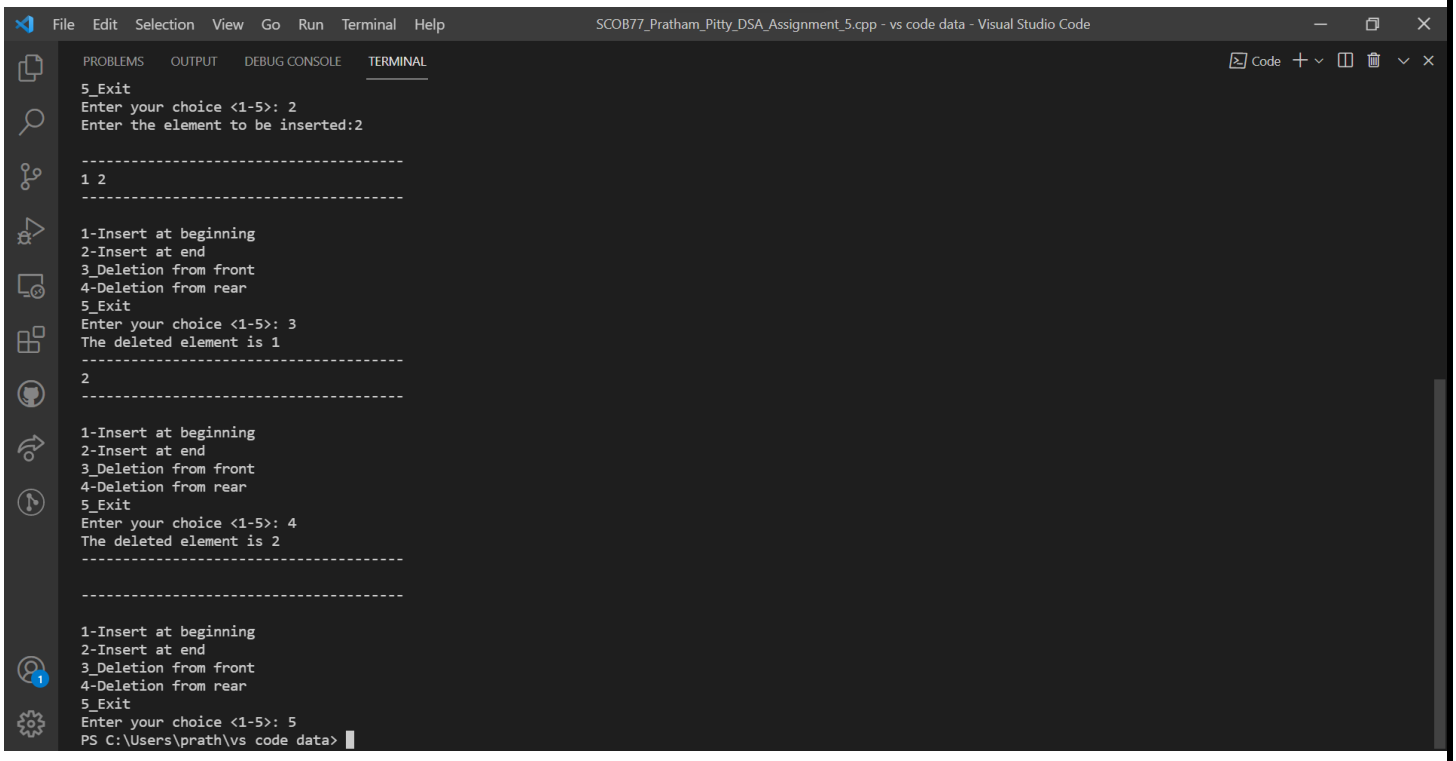
Output :-


File Edit Selection View Go Run Terminal Help
SCOB77_Pratham_Pitty_DSA_Assignment_5.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
PS C:\Users\prath\vs code data> cd "c:\Users\prath\vs code data\" ; if ($?) { g++ SCOB77_Pratham_Pitty_DSA_Assignment_5.cpp -o SCOB77_Pratham_Pitty_DSA_Assignment_5 } ; if ($?) { .\SCOB77_Pratham_Pitty_DSA_Assignment_5 }
****DEQUEUE OPERATION****
1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 1
Enter the element to be inserted:1

1

1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 2
Enter the element to be inserted:2
---
1 2

1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 3
The deleted element is 1


File Edit Selection View Go Run Terminal Help
SCOB77_Pratham_Pitty_DSA_Assignment_5.cpp - vs code data - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
5_Exit
Enter your choice <1-5>: 2
Enter the element to be inserted:2
---
1 2

1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 3
The deleted element is 1

2

1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 4
The deleted element is 2

1-Insert at beginning
2-Insert at end
3_Deletion from front
4-Deletion from rear
5_Exit
Enter your choice <1-5>: 5
PS C:\Users\prath\vs code data>