NAME:SAYALI JIVAN CHAUDHARI

ROLL NO.:14

PRN NO.2023015400005055

1)implementation of operation based on dequeue

```
#include<iostream>
using namespace std;
#define SIZE 10
class dequeue {
   int a[20],f,r;
   public:
      dequeue();
      void insert_at_beg(int);
      void insert_at_end(int);
      void delete_fr_front();
      void delete_fr_rear();
      void show();
};
dequeue::dequeue() {
```

```
  f=-1;
  r=-1;
}
void dequeue::insert_at_end(int i) {
  if(r>=SIZE-1) {
    cout<<"\n insertion is not possible, overflow!!!!";
  } else {
    if(f==-1) {
      f++;
      r++;
    } else {
      r=r+1;
    }
    a[r]=i;
    cout<<"\nInserted item is"<<a[r];
  }
}
void dequeue::insert_at_beg(int i) {
  if(f==-1) {
```

```
      f=0;
      a[++r]=i;
      cout<<"\n inserted element is:"<<i;
    } else if(f!=0) {
      a[--f]=i;
      cout<<"\n inserted element is:"<<i;
    } else {
      cout<<"\n insertion is not possible, overflow!!!";
    }
}
void dequeue::delete_fr_front() {
    if(f==-1) {
      cout<<"deletion is not possible::dequeue is empty";
      return;
    }
    else {
      cout<<"the deleted element is:"<<a[f];
      if(f==r) {
        f=r=-1;
```

```
    return;
  } else
    f=f+1;
  }
}
void dequeue::delete_fr_rear() {
  if(f==-1) {
    cout<<"deletion is not possible::dequeue is empty";
    return;
  }
  else {
    cout<<"the deleted element is:"<<a[r];
    if(f==r) {
      f=r=-1;
    } else
      r=r-1;
  }
}
void dequeue::show() {
```

```
  if(f==-1) {
    cout<<"Dequeue is empty";
  } else {
    for(int i=f;i<=r;i++) {
      cout<<a[i]<<" ";
    }
  }
}
int main() {
  int c,i;
  dequeue d;
  Do//perform switch opeartion {
  cout<<"\n 1.insert at beginning";
  cout<<"\n 2.insert at end";
  cout<<"\n 3.show";
  cout<<"\n 4.deletion from front";
  cout<<"\n 5.deletion from rear";
  cout<<"\n 6.exit";
  cout<<"\n enter your choice:";
```

```
cin>>c;
switch(c) {
  case 1:
    cout<<"enter the element to be inserted";
    cin>>i;
    d.insert_at_beg(i);
  break;
  case 2:
    cout<<"enter the element to be inserted";
    cin>>i;
    d.insert_at_end(i);
  break;
  case 3:
    d.show();
  break;
  case 4:
    d.delete_fr_front();
  break;
  case 5:
```

```
        d.delete_fr_rear();
      break;
      case 6:
        exit(1);
      break;
      default:
        cout<<"invalid choice";
      break;
    }
  } while(c!=7);
}
```