**Name # Noman Amjad**

**Class # bscs 3b**

**Reg No # 21-arid-654**

**Code 1 :**

**//Stact implementation using array;**

**#include<iostream>**

**using namespace std;**

**class stack**

**{**

**int arr[5];**

**int top;**

**public:**

**stact()**

**{**

**top=-1;**

**}**

**void push()**

**{**

**int val;**

**cout<<"Enter a value :";**

**cin>>val;**

**if(top<4)**

**{**

**arr[++top]=val;**

**}**

**else**

**{**

**cout<<"Stack is full..."<<endl;**

**}**

**}**

**int pop()**

**{**

**if(top>-1)**

**{**

**return arr[top--];**

**}**

**else**

**cout<<"Stack is Empty..."<<endl;**

**}**

**};**

**int main()**

**{**

**stack s;**

**for(int i=0;i<5;i++)**

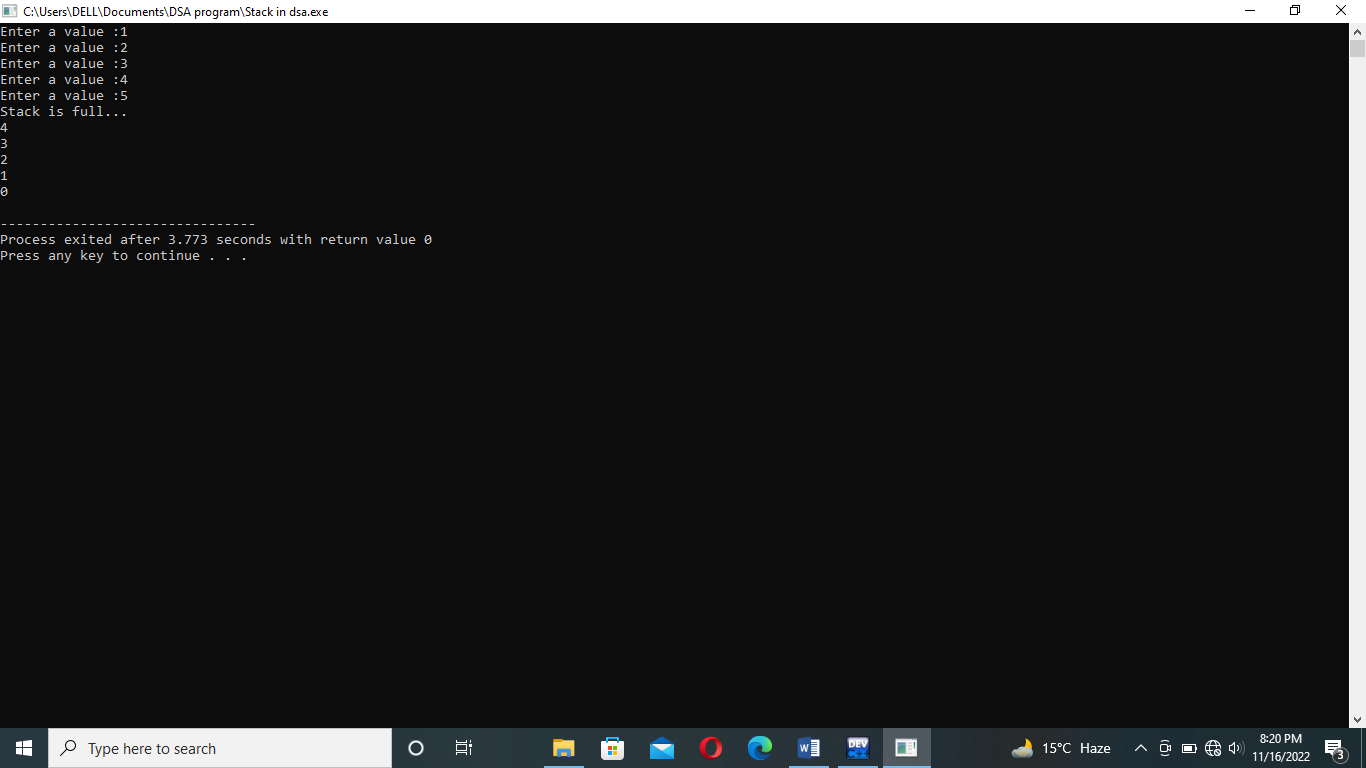
**s.push();**

**for(int i=0;i<5;i++)**

**cout<<s.pop()<<endl;**

**}**

**Output:**

****

**2)Stack implementation using link list insertion at front and deletion from front ::**

**Code 2:**

**#include<iostream>**

**using namespace std;**

**class Stack**

**{**

**public:**

**int number;**

**Stack \*next;**

**Stack \*start=NULL;**

**void InsertInFront()**

**{**

**Stack \*temp = new Stack();**

**cout<<"Enter number : ";**

**cin>>temp->number;**

**temp->next = start;**

**start = temp;**

**}**

**void DeleteInFront()**

**{**

**if(start==NULL)**

**{**

**cout<<"No node exists";**

**}**

**else**

**{**

**Stack \*temp=start;**

**Stack \*nodeDel = temp;**

**start = temp->next;**

**delete nodeDel;**

**}**

**}**

**void Display()**

**{**

**if(start == NULL)**

**{**

**cout<<"No node exists";**

**}**

**else**

**{**

**Stack \*temp = start;**

**cout<<"\*\*\*\*Stack elements\*\*\*\*\n";**

**while(temp!=NULL)**

**{**

**cout<<temp->number<<endl;**

**temp=temp->next;**

**}**

**}**

**}**

**};**

**int main()**

**{**

**Stack obj;**

**int opt;**

**do**

**{**

**int option;**

**cout<<"\n---Stack using link list\n";**

**cout<<"Enter 1 to insert in start\n";**

**cout<<"Enter 2 to delete from start\n";**

**cout<<"Enter 3 to display stack elements\n";**

**cout<<"Choice : ";**

**cin>>option;**

**switch(option)**

**{**

**case 1:**

**obj.InsertInFront();**

**break;**

**case 2:**

**obj.DeleteInFront();**

**break;**

**case 3:**

**obj.Display();**

**break;**

**default:**

**cout<<"Invalid option";**

**break;**

**}**

**cout<<"Press 1 to continue : ";**

**cin>>opt;**

**}while(opt==1);**

**}**

Code 3:

**#include<iostream>**

**using namespace std;**

**class Stack**

**{**

**public:**

**int number;**

**Stack \*next;**

**Stack \*start=NULL;**

**void InsertAtEnd()**

**{**

**if(start == NULL)**

**{**

**Stack \*temp = new Stack();**

**cout<<"Enter number : ";**

**cin>>temp->number;**

**start = temp;**

**}**

**else**

**{**

**Stack \*temp = start;**

**while(temp->next != NULL)**

**{**

**temp = temp->next;**

**}**

**Stack \*t = new Stack();**

**cout<<"Enter number : ";**

**cin>>t->number;**

**temp->next = t;**

**}**

**}**

**void DeleteFromEnd()**

**{**

**if(start==NULL)**

**{**

**cout<<"No node exists\n";**

**}**

**else if(start->next==NULL)**

**{**

**Stack \*nodeDel = start;**

**start = NULL;**

**delete nodeDel;**

**}**

**else**

**{**

**Stack \*temp = start;**

**while(temp->next->next != NULL)**

**{**

**temp = temp->next;**

**}**

**Stack \*nodeDel = temp->next;**

**temp->next = NULL;**

**delete nodeDel;**

**}**

**}**

**int Display(Stack \*st)**

**{**

**if(st==NULL)**

**{**

**return 1;**

**}**

**else**

**{**

**Display(st->next);**

**cout<<st->number<<endl;**

**}**

**}**

**};**

**int main()**

**{**

**Stack obj;**

**int opt;**

**do**

**{**

**int option;**

**cout<<"\n---Stack using link list---\n";**

**cout<<"Enter 1 to insert in end\n";**

**cout<<"Enter 2 to delete from end\n";**

**cout<<"Enter 3 to display stack elements\n";**

**cout<<"Choice : ";**

**cin>>option;**

**switch(option)**

**{**

**case 1:**

**obj.InsertAtEnd();**

**break;**

**case 2:**

**obj.DeleteFromEnd();**

**break;**

**case 3:**

**obj.Display(obj.start);**

**break;**

**default:**

**cout<<"Invalid option";**

**break;**

**}**

**cout<<"Press 1 to continue : ";**

**cin>>opt;**

**}while(opt==1);**

**}**

**Code 4:**

**#include<iostream>**

**using namespace std;**

**class queue**

**{**

**int que[5];**

**int n;**

**int front;**

**int rear;**

**public:**

**queue()**

**{**

**front=-1;**

**rear=-1;**

**n=5;**

**}**

**void insert()**

**{**

**int v;**

**if(rear==n-1)**

**{**

**cout<<"\nQueue is Overflow\n";**

**}**

**else**

**{**

**if(front==-1)**

**front=0;**

**cout<<"\nEnter a Value please : ";**

**cin>>v;**

**rear++;**

**que[rear]=v;**

**}**

**}**

**void deleted()**

**{**

**if(front==-1 || front>rear)**

**{**

**cout<<"\nQueue is Underflow\n";**

**}**

**else**

**{**

**cout<<"\nElement "<<que[front]<<" Deleted...!!"<<endl;**

**front++;**

**}**

**}**

**void display()**

**{**

**if(front==-1)**

**{**

**cout<<"\nQueue IS Empty...\n";**

**}**

**else**

**{**

**for(int i=0;i<rear+1;i++)**

**{**

**cout<<que[i]<<endl;**

**}**

**}**

**}**

**};**

**int main()**

**{**

**queue q;**

**cout<<"Enter 5 Array element : ";**

**for(int i=0;i<6;i++)**

**{**

**q.insert();**

**}**

**// cout<<"\nArray element is :\n";**

**// q.display();**

**cout<<"\nDELETE element Using QUEUE method :\n";**

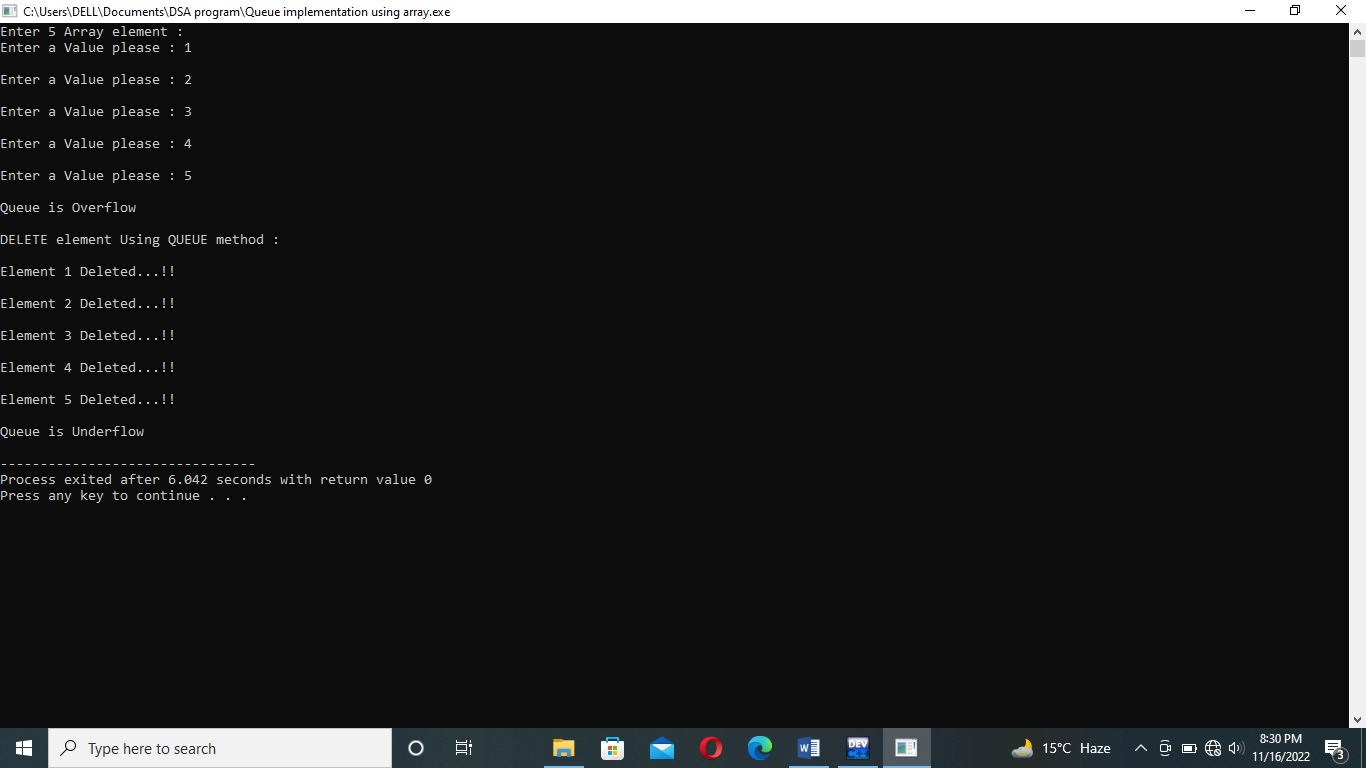
**for(int j=0;j<6;j++)**

**q.deleted();**

**return 0;**

**}**

**Output:**

****

**CODE 5 and 6:---------------------------------------------------------------------------------------------**

**#include<iostream>**

**using namespace std;**

**class Queue;**

**Queue \*start=NULL;**

**class Queue**

**{**

**int num;**

**Queue \*next;**

**public:**

**void En\_QUEUE\_FROM\_FRONT()**

**{**

**Queue \*t=new Queue();**

**cout<<"\nEnter a Number Please : ";**

**cin>>t->num;**

**t->next=start;**

**start=t;**

**}**

**void De\_QUEUE\_FROM\_END()**

**{**

**if(start==NULL)**

**{**

**cout<<"\nQueue is Empty No Node exists...!!\n";**

**}**

**else if(start->next==NULL)**

**{**

**Queue \*d=start;**

**start=NULL;**

**delete d;**

**cout<<"\nNode Deleted Successfully From End...\n";**

**}**

**else**

**{**

**Queue \*d=start;**

**while(d->next->next!=NULL)**

**{**

**d=d->next;**

**}**

**d->next=NULL;**

**cout<<"\nNode is DELETED Successfully From End....\n";**

**}**

**}**

**void EnQueueFromENd()**

**{**

**Queue \*t=new Queue();**

**if(start==NULL)**

**{**

**cout<<"\nQueue is Empty So Insert Node at Here; \n";**

**cout<<"\nEnter a NUMber :";**

**cin>>t->num;**

**cout<<"\nNode Inserted succussfully...\n";**

**}**

**else**

**{**

**Queue \*s=start;**

**cout<<"\nEnter A NUMBER : ";**

**cin>>t->num;**

**while(s->next!=NULL)**

**{**

**s=s->next;**

**}**

**s->next=s;**

**cout<<"\nNOde inserted succussfully at END ..\n";**

**}**

**}**

**void display()**

**{**

**if(start==NULL)**

**{**

**cout<<"\nNO Node exists !!!\n";**

**}**

**else**

**{**

**Queue \*s=start;**

**cout<<"\nData in Queue :\n";**

**while(s!=NULL)**

**{**

**cout<<s->num<<endl;**

**s=s->next;**

**}**

**}**

**}**

**void deQUEUEfromFront()**

**{**

**if(start==NULL)**

**{**

**cout<<"No Node to Deleted Empty Queue....\n";**

**}**

**else**

**{**

**Queue \*T2=start;**

**start=start->next;**

**delete T2;**

**cout<<"\nNode is deleted succussfullay.......\n";**

**}**

**}**

**};**

**int main()**

**{**

**Queue q;**

**int choice ;**

**do{**

**cout<<"\n1) FOR En Queue From Front :";**

**cout<<"\n2) FOR De Queue From END :";**

**cout<<"\n3) For Display : ";**

**cout<<"\n4) For En Queue From END : ";**

**cout<<"\n5) For De Queue from Front : ";**

**cout<<"\nEnter a choice : ";**

**cin>>choice;**

**switch(choice)**

**{**

**case 1:**

**q.En\_QUEUE\_FROM\_FRONT();**

**break;**

**case 2:**

**q.De\_QUEUE\_FROM\_END();**

**break;**

**case 3:**

**q.display();**

**break;**

**case 4:**

**q.EnQueueFromENd();**

**break;**

**case 5:**

**q.deQUEUEfromFront();**

**break;**

**case 0:**

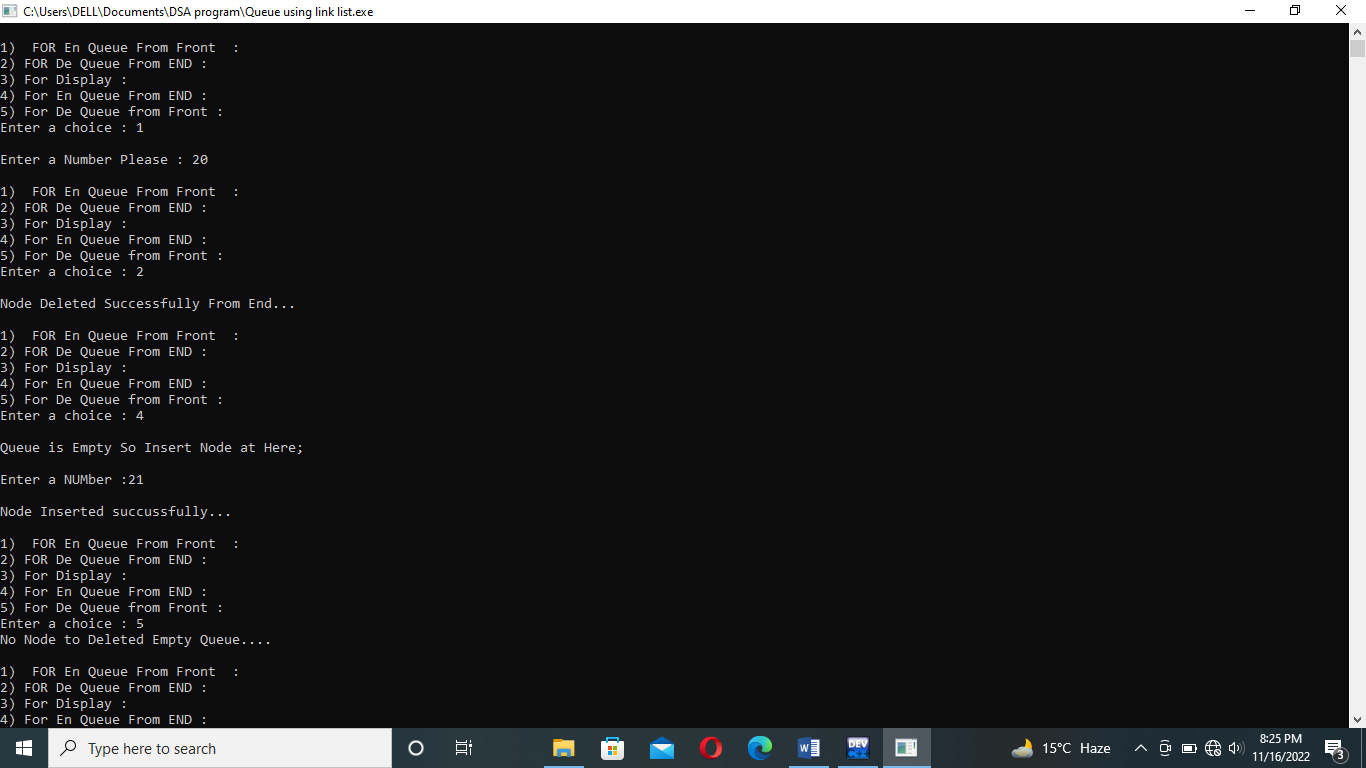
**exit(1);**

**}**

**}while(choice!=0);**

**}**

**Output:**

****