**Debre Markos University**

**Department of Information Technology**

**Data Structure and Algorithm Lab Manual**

**Lab 4: Array implementation of Stack and Queue operations**

**Tools used:** Quincy 2005v.1.3 editor

**Objective:**

Students should be able to know:

- What push and pop operations of stack are, and their implementation using array

- What enqueue and dequeue operations of queue are, and their implementation using array

**Source code for array implementation of Stack operations (PUSH and POP )**

#include<iostream.h>

#define size 10

int Stack[size];

int top = -1;

void push(int x)

{

if(top = = size)

cout<<"\n Stack is full."<<endl;

else

{

top++;

Stack[top] = x;

} }

int pop()

{

if(top = = -1)

{

cout<<"\n Stack is empty."<<endl;

return -1;

}

int t = Stack[top];

top--;

return t;

}

void display()

{

int t=top;

while(t != -1)

{

cout<<Stack[t]<<endl;

t--;

}

}

int main()

{

int s;

int a[10];

cout<<"How many elements do u want to store in a stack (<10):";

cin>>s;

cout<<"\n\t Enter:\n";

for(int i=0;i<s; i ++)

{

cout<<"\t\t element "<<(i+1)<<" :";

cin>>a[i];

push(a[i]);

}

cout<<"\n The stack is:"<<endl;

display();

int t;

for(int k=1;k<=s; k ++)

{

t=pop();

if(k= =1)

cout<<"1st popped element is : "<<t<<endl;

else if(k==2)

cout<<"2nd popped element is : "<<t<<endl;

else if(k==3)

cout<<"3rd popped element is : "<<t<<endl;

else

cout<<k<<"th popped element is : "<<t<<endl;

}

}

**Page - 1**

**Source code for array implementation of Queue operations (ENQUEUE and DEQUEUE)**

#include<iostream.h>

#define size 10

int queue[size];

int front = -1, rear = -1;

int isempty()

{

return front = = -1;

}

int isfull()

{return ((rear = = size-1));

}

void enqueue(int item)

{

if(isfull())

cout<<"Overflow, Queue is full.\n";

else

{

queue[++rear]=item;

if(front = = -1)

front++;

}

}

void display()

{int r, f;

r =rear;

f=front;

while(f != r+1)

{cout<<queue[f]<<" ";

f=f+1;

}

cout<<endl;

}

int dequeue()

{

if(isempty())

{

cout<<"Underflow, Queue is empty.\n";

return NULL;

}

else

return queue[front++];

}

**Page - 2**

int main()

{

int a[size];

int s;

cout<<"How many elements do you have for queue?(<10):";

cin>>s;

cout<<"\n Enter...\n";

for(int i=0;i<s; i++)

{

cout<<"\t element "<<(i+1)<<" :";

cin>>a[i];

enqueue(a[i]);

}

cout<<"\n The queue looks like:\n";

display();

cout<<endl;

int t;

for(int k = 1;k <= s; k++)

{

t=dequeue();

if(k = = 1)

cout<<"1st deleted item is: "<<t<<endl;

else if(k = = 2)

cout<<"2nd deleted item is: "<<t<<endl;

else if(k = = 3)

cout<<"3rd deleted item is: "<<t<<endl;

else

cout<<k<<"th deleted item is: "<<t<<endl;

}

}