**MANGALORE UNIVERSITY**

**Bachelor of Computer Applications (BCA) Degree Programme**

**Choice Based Credit System (2019-2020 Onwards)**

**III Semester – Practicals**

**BCAP 234**

**Operating Systems and Data Structures Lab**

**Part - A:**

Implementations using C++

1. **Write a program to demonstrate binary search.**

#include<iostream.h>

#include<iomanip.h>

#include<conio.h>

int main()

{

int n,a[10],i,beg=0,end ,item,mid,flag=0;

clrscr();

cout<<"Enter the no of elements: ";

cin>>n;

end=n-1;

cout<<"Enter the sorted array elements: \n";

for(i=0;i<n;i++)

{

cin>>a[i];

}

cout<<"Enter the array element to be searched "<<endl;

cin>>item;

mid=(beg+end)/2;

for(;beg<=end&&a[mid]!=item;)

{

if(item<a[mid])

end=mid-1;

else

beg=mid+1;

mid=(beg+end)/2;

}

if(a[mid]==item)

{

flag=1;

}

if(flag==0){

cout<<"Search unsccessful"<<endl;

}

else

cout<<"search is successful and found at position "<<mid+1<<endl;

getch();

return 0;

}

/\*

**Output:**

Enter the no of elements: 4

Enter the sorted array elements:

1

2

3

4

Enter the array element to be searched

3

search is successful and found at position 3

Enter the no of elements: 3

Enter the sorted array elements:

1

2

3

Enter the array element to be searched

5

Search unsccessful

\*/

1. **Write a program to demonstrate merge sort.**

#include<iostream.h>

#include<conio.h>

void mergesort(int a[10],int,int);

void simplemerge(int a[10],int,int,int);

int main()

{

int a[10],i,n,mid,low,high;

clrscr();

cout<<"Enter the number of elements:"<<endl;

cin>>n;

cout<<"Enter the array elements:"<<endl;

for(i=0;i<n;i++)

{

cin>>a[i];

}

low=0;

high=n-1;

mergesort(a,low,high);

cout<<"Sorted array is : "<<endl;

for(i=0;i<n;i++)

cout<<a[i]<<" ";

cout<<endl;

getch();

return 0;

}

void mergesort(int a[10],int low,int high)

{

int mid;

if(low!=high)

{

mid=(low+high) /2;

mergesort(a,low,mid);

mergesort(a,mid+1,high);

simplemerge(a,low,mid,high);

}

}

void simplemerge(int a[10],int low,int mid,int high)

{

int c[20],k,i,j;

i=low;

j=mid+1;

k=low;

while(i<=mid && j<=high)

{

if(a[i]<=a[j])

{

c[k]=a[i];

i=i+1;

k=k+1;

}

else

{

c[k]=a[j];

j=j+1;

k=k+1;

}

}

while(i<=mid)

{

c[k]=a[i];

i=i+1;

k=k+1;

}

while(j<=high)

{

c[k]=a[j];

j=j+1;

k+=1;

}

for(i=low;i<=high;i++)

{

a[i]=c[i];

}

}

/\*

**output:**

Enter the number of elements: 5

Enter the array elements:

1

5

-1

20

50

Sorted elements are

-1 1 5 20 50

\*/

1. **Write a program to demonstrate insertion sort.**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

int main()

{

clrscr();

int ptr,temp,i,n,a[10];

cout<<"Enter the number of elements: ";

cin>>n;

cout<<"Enter the array elements:"<<endl;

for(i=0;i<n;i++)

{

cin>>a[i];

}

for(i=0;i<n;i++)

{

temp=a[i];

ptr=i-1;

while(temp<a[ptr]&&ptr>=0)

{

a[ptr+1]=a[ptr];

ptr=ptr-1;

}

a[ptr+1]=temp;

}

cout<<"Sorted elements are"<<endl;

for(i=0;i<n;i++)

{

cout<<a[i]<<" ";

}

getch();

return 0;

}

/\*

**Output:**

Enter the number of elements: 5

Enter the array elements:

1

4

-1

2

100

Sorted elements are

-1 1 2 4 100

Enter the number of elements: 5

Enter the array elements:

1

5

-1

20

50

Sorted elements are

-1 1 5 20 50

\*/

1. **Write a program to implement queue using arrays.**

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

#define max 3

int f=-1;

int r=-1;

int q[max];

void qinsert();

void qdelete();

void qdisplay();

void main()

{

int abc=1,ch;

clrscr();

while(abc)

{

cout<<"Queue"<<endl;

cout<<"1.Queue insert"<<endl;

cout<<"2.Queue delete"<<endl;

cout<<"3.Queue display"<<endl;

cout<<"Enter your choice"<<endl;

cin>>ch;

switch(ch)

{

case 1:

qinsert();

break;

case 2:

qdelete();

break;

case 3:

qdisplay();

break;

default : return;

}

cout<<"Do you want to continue? press 1 or 0"<<endl;

cin>>abc;

}

getch();

}

void qinsert(){

int x;

cout<<"enter a number "<<endl;

cin>>x;

if(r>max-1)

{

cout<<"overflow"<<endl;

return;

}

else

{

r=r+1;

q[r]=x;

}

if(f==-1)

{

f=f+1;

}

}

void qdelete()

{

int x;

if(f==-1)

{

cout<<"Underflow"<<endl;

}

else

{

if(f==r)

{

f=-1;

x=q[r];

cout<<"deleted element is "<<x<<endl;

}

else

{

x=q[f];

f++;

cout<<"the deleted element is "<< x<<endl;

}

if(f>max-1)

{

f=-1;

}

}

}

void qdisplay()

{

int i;

if(f==-1)

{

cout<<"Queue is empty"<<endl;

}

else

{

cout<<"queue elements are: "<<endl;

for(i=f;i<=r;i++)

cout<<q[i]<<endl;

}

}

/\*

**Output:**

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

1

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

2

deleted element is 1

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

3

Queue is empty

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

1

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

2

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

3

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

4

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

1

enter a number

5

overflow

Do you want to continue? press 1 or 0

1

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

3

queue elements are:

1

2

3

4

Do you want to continue? press 1 or 0

0

Queue

1.Queue insert

2.Queue delete

3.Queue display

Enter your choice

2

Underflow

Do you want to continue? press 1 or 0

0

\*/

1. **Write a program to implement stack using arrays.**

#include<iostream.h>

#include<process.h>

#include<conio.h>

#include<math.h>

#define MAX 3

class Stack

{

int top;

int a[MAX];

public:

Stack()

{

top=-1;

}

void push(int x);

void pop();

void display();

int stackfull();

int stackempty();

};

void Stack::push(int x)

{

if(stackfull())

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

else

a[++top]=x;

}

void Stack :: pop()

{

int c;

if(stackempty())

cout<<"Stack is empty"<<endl;

else{

c=a[top--];

cout<<"Poped element is "<<c<<endl;

}

}

void Stack::display()

{

int i;

if (top==-1)

cout<<"Stack is empty"<<endl;

else{

cout<<"Stack elements are: "<<endl;

for(i=top;i>=0;i--){

cout<<a[i]<<endl;

}

}

cout<<endl;

}

int Stack:: stackfull()

{

if(top==(MAX-1))

return 1;

else

return 0;

}

int Stack::stackempty()

{

if(top==-1)

return 1;

else

return 0;

}

int main()

{

int ch,t;

Stack s;

clrscr();

do

{

cout<<"\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"1.Push"<<endl;

cout<<"2.Pop"<<endl;

cout<<"3.Display"<<endl;

cout<<"4.Exit"<<endl;

cout<<"Enter your choice: ";

cin>>ch;

switch(ch)

{

case 1:cout<<"Enter the element to insert: "<<endl;

cin>>t;

s.push(t);

break;

case 2:s.pop();

break;

case 3:

s.display();

break;

case 4:exit(0);

}

}while(ch>=1 && ch<=3);

getch();

return 0;

}

/\*

**Output:**

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Output:

Enter the element to insert:

1

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Enter the element to insert:

2

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Enter the element to insert:

3

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Enter the element to insert:

4

Stack is full

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 3

Stack elements are:

3

2

1

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 2

Poped element is 3

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 3

Stack elements are:

2

1

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 4

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 2

Stack is empty

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 3

Stack is empty

\*\*\*\*\*\*\*\*\*\*

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 4

\*/