1.1Linear_Search_13

Code:-

```cpp
#include<iostream>

using namespace std;

class Nick

{

private:

int a[100],num,i,j,temp,p;

public:

void getData() {

cout<<"Enter the size of array : ";

cin>>num;

cout<<"Enter "<<num<<" Elements : ";

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

{

cin>>a[i];

}

}

void showData()

{

cout<<"Array Elements : ";

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

{

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

}

cout<<"\n";

}

void linearSearch()

{

int no,flag,x;

flag = 0;
```

Searching and hashing

```
cout<<"\nEnter the element which you want to search ? ";

cin>>no;

for(x=0;x<num;x++)

{

if(no==a[x])

{

flag = 1;

cout<<"\nEntered Element "<<no<<" Found at "<<x+1<<" with index no "<<x<<endl;

break;

}

}

if(flag==0)

{

cout<<"\nEnterd Value Not Found...array exausted &Added to the End.\n";

a[x]=no;

showData();

}

}

};

int main()

{

Nick n;

n.getData();

n.linearSearch();

return 0;

}
```

Output:-

Searching and hashing

```
Enter the size of array : 5
Enter 5 Elements : 1
2
3
4
5

Enter the element which you want to search ? 3

Entered Element 3 Found at 3 with index no 2
```

1.2binary_Search_13

Code:-

#include<iostream>

using namespace std;

class Nick{

private:

int a[100],num,i,j,temp;

public:

void getData()

{

cout<<"Enter the size of array : ";

cin>>num;

cout<<"Enter "<<num<<" Elements : ";

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

{

cin>>a[i];

}

}

void showData() {

cout<<"Array Elements : ";

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

{

Searching and hashing

```
cout<<"\t"<<a[i];

}

cout<<"\n";

}

void binarySearch() {

int no,flag,k,low,high,mid;

flag = 0;

cout<<"\nEnter the element/number which you want to search ? ";

cin>>no;

low=0;

high=num-1;

do

{

mid=(low+high)/2;

if(no==a[mid])

{

flag=1;

cout<<"\nEntered Element "<<no<<" Found at "<<mid+1<<" with index no "<<mid<<endl;

break;

}

if(no>a[mid])

{

low=mid+1;

}

if(no<a[mid])

{

high=mid-1;

}

}

while(low<=high);

if(flag==0)
```

Searching and hashing

```
{
cout<<"\nEnterd Value Not Found.array exausted & Added in the End.\n";
}
}
void bubbleSort()
{
for(int i=0;i<(num-1);i++)
{
for(j=0;j<(num-i)-1;j++)
{
if(a[j] > a[j+1])
{
a[j]>a[j+1];
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
cout<<"\t** After Sorting **\n";
showData();
}
};
int main()
{
Nick n;
n.getData();
n.bubbleSort();
n.binarySearch();
return 0;
}
```

Searching and hashing

Output:-

1.3Modulo_Division_Hashing_13

Code:-

```
#include<iostream>

using namespace std;

class nick{

private:

int n, s,*a,num,add;

public:

void get()

{

do

{

cout << "Enter array size:-";

cin >> s;

cout << "how many numbers you wanna insert:-";

cin >> n;

}
while(n>s);

a=new int[s];

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

Searching and hashing

```
a[i] = '\0';

}

void mytech()

{

cout << endl;

while(n--)

{

cout << "Enter numbers:-";

cin >> num;

if (num < 0)

{

cout << "Invalid number.Try Again\n";

n++;continue;

}

add = num % s;

lp();

a[add] =  num;

}

dis();

}

void lp(){

while(a[add]!='\0'){

if(add < s-2)

{

add++;

}

else

{

add = 0;

}

}
```

Searching and hashing

```
}
void dis()
{
cout << "\nhashed array is as below\n";
for(int i = 0; i < s; i++){
cout << "array index:-" << i << " : \t";
if(a[i]=='\0')
{
cout << "NULL" << endl;
}
else
{
cout << a[i] << endl;
}
}
};
int main()
{
nick o;
o.get();
o.mytech();
}
```

Output:-

Searching and hashing

```
Enter array size:-10
how many numbers you wanna insert:-5

Enter numbers:-1
Enter numbers:-2
Enter numbers:-3
Enter numbers:-4
Enter numbers:-5

hashed array is as below
array index:-0 :          NULL
array index:-1 :          1
array index:-2 :          2
array index:-3 :          3
array index:-4 :          4
array index:-5 :          5
array index:-6 :          NULL
array index:-7 :          NULL
array index:-8 :          NULL
array index:-9 :          NULL
```

1.4 Digit_Extraction_Hashing_13

Code:-

#include<iostream>

#include<math.h>

using namespace std;

class nick{

private:

int n,s,*a,num,add,*loc,no;

public:

void get()

{

do

{

cout << "Enter the size of the array: ";

cin >> s;

cout << "Enter the no. of digits you want to insert: ";

Searching and hashing

```
cin >> n;

}

while(n>s);

a=new int[s];

cout<<"\nEnter the no. of digits you want to extract: ";

cin>>no;

loc = new int[no];

for(int i=0;i<no;i++)

{

cout<<"Enter location no. "<<i+1<<" you want to extract: ";

cin >> loc[i];

}

sort();

a= new int[s];

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

{

a[i]=-1;

}

}

void mytech()

{

cout<<"\n";

while(n--)

{

add= 0;

cout<<"Enter value you want to insert: ";

cin>>num;

if(num < 0)

{

cout<< "Invalid Input. Try Again\n";

n++;continue;
```

Searching and hashing

```
}

for(int i=0;i< no; i++)

{

num+= ((num% int(pow(10.0,float(loc[i])))) / (int(pow(10.0,float(loc[i])))/10)) *
int(pow(10.0,float(i)));

}

if(add>=s)

{

add=add%s;

}

lp();

a[add]=num;

}

dis();

}

void lp()

{

while(a[add]!=-1)

{

if(add<s-2)

{

add++;

}

else

{

add=0;

}

}

}

void sort()

{
```

Searching and hashing

```
for(int i=0;i<no;i++)

{

for(int j=0;j<(no-i)-1;j++)

{

if(a[j]>a[j+1])

{

int t = a[j];

a[j] = a[j+1];

a[j+1]=t;

}

}

}

}

void dis()

{

cout << "\nhashed array is as below\n";

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

{

cout << "Index no. " << i <<" : \t";

if(a[i]==-1)

cout<<"NULL"<<endl;

else

{

cout << a[i] << endl;

}

}

}

};

int main()

{

nick o;
```

Searching and hashing

o.get();

o.mytech();

}

Output:-

```
Enter the size of the array: 10
Enter the no. of digits you want to insert: 5

Enter the no. of digits you want to extract: 3
Enter location no. 1 you want to extract: 1
Enter location no. 2 you want to extract: 2
Enter location no. 3 you want to extract: 4

Enter value you want to insert: 12345
Enter value you want to insert: 65432
Enter value you want to insert: 76345
Enter value you want to insert: 23413
Enter value you want to insert: 98756

hashed array is as below
Index no. 0 :    12600
Index no. 1 :    65964
Index no. 2 :    77000
Index no. 3 :    23726
Index no. 4 :    99622
Index no. 5 :    NULL
Index no. 6 :    NULL
Index no. 7 :    NULL
Index no. 8 :    NULL
Index no. 9 :    NULL
```

1.5 Mid_square_Hashing_13

Code:-

```cpp
#include<iostream>

#include<math.h>

using namespace std;

class nick{

private:

int n, s,*a,num,add;

public:
```

Searching and hashing

```
void get()

{

do

{

cout << "Enter array size:-";

cin >> s;

cout << "how many numbers you wanna insert:-";

cin >> n;

}

while(n>s);

a=new int[s];

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

a[i] = '\0';

}

void mytech()

{

cout << endl;

while(n--)

{

cout << "Enter number:- ";

cin >> num;

if (num < 0)

{

cout << "Invalid value. Try Again\n";

n++;continue;

}

add = num*num;

while(int(log10(add)+1)>2)

{

add /= 10;

add %= int(pow(10.0,float(int(log10(add)))));
```

Searching and hashing

```
if(add<s)

{

break;

}

}

if(add>=s)

add=add%s;

lp();

a[add] = num;

}

dis();

}

void lp(){

while(a[add]!='\0'){

if(add < s-2)

{

add++;

}

else

{

add = 0;

}

}

}

void dis()

{

cout << "\nhashed array is as below\n";

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

cout << "array index:-" << i << " : \t";

if(a[i]=='\0')

{
```
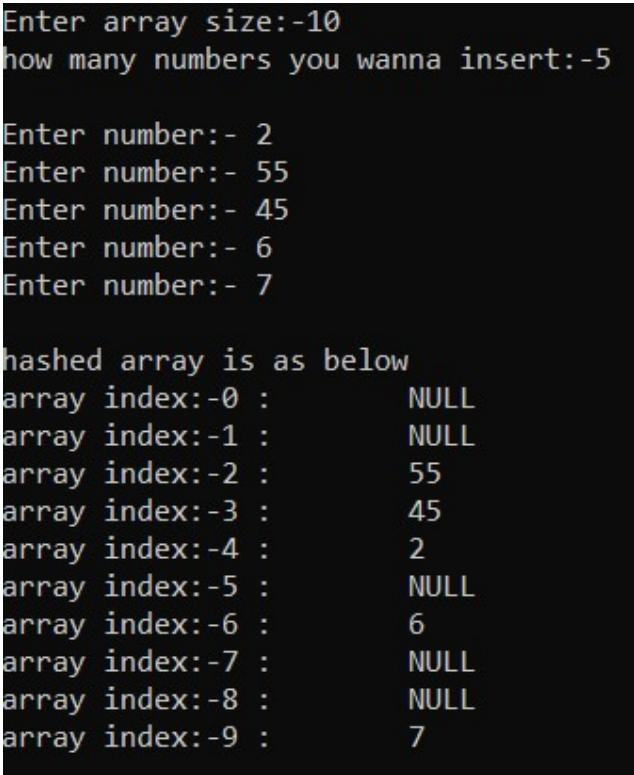
Searching and hashing

```
cout << "NULL" << endl;

}

else

{

cout << a[i] << endl;

}

}

}

};

int main()

{

nick o;

o.get();

o.mytech();

}
```

Output:-

```
Enter array size:-10
how many numbers you wanna insert:-5

Enter number:- 2
Enter number:- 55
Enter number:- 45
Enter number:- 6
Enter number:- 7

hashed array is as below
array index:-0 :          NULL
array index:-1 :          NULL
array index:-2 :          55
array index:-3 :          45
array index:-4 :          2
array index:-5 :          NULL
array index:-6 :          6
array index:-7 :          NULL
array index:-8 :          NULL
array index:-9 :          7
```

Searching and hashing

1.6Folding_Boundary_Hashing_13

Code:-

```
#include<iostream>

#include<math.h>

using namespace std;

class nick

{

private:

int n,s,*a,num;

unsigned long int add;

public:

void get()

{

do

{

cout<<"Enter the size of the array: ";

cin>>s;

cout<<"Enter the no. of digits you want to insert: ";

cin>>n;

}

while(n>s);

a=new int[s];

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

{

a[i]=-1;

}

}

void mytech()

{

cout<<"\n";

while(n--)
```

Searching and hashing

```
{
cout<<"Enter value you want to insert: ";
cin>>num;
if(num< 0)
{
cout<<"Invalid value. Try Again\n";
n++;continue;
}
int t=num;
add=0;
int round = 0;
while(t>0)
{
if((round==0) || (t<s))
{
add+=rev(t%s);
}
else
{
add+= t%s;
}
t/=s;
round++;
}
add%=100;
if(add>=s)
{
add=add%s;
}
lp();
a[add]=num;
```

Searching and hashing

```
}
dis();
}
void lp()
{
while(a[add]!=-1)
{
if(add<s-2)
{
add++;
}
else
{
add=0;
}
}
}
int rev(int no)
{
if(no<10)
{
return (no*10);
}
else
{
int reverse = 0;
while(no>0){
reverse = (reverse*10)+(no%10);
no/=10;
}
return reverse;
```

Searching and hashing

```cpp
}

}

void dis()

{

cout << "\nhashed array is as below\n";

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

{

cout << "array Index " << i << " : \t";

if(a[i]==-1)

{

cout << "NULL" << endl;

}

else

{

cout << a[i] << endl;

}

}

}

};

int main()

{

nick o;

o.get();

o.mytech();

}
```

Output:-

Searching and hashing

```
Enter the size of the array: 10
Enter the no. of digits you want to insert: 5

Enter value you want to insert: 234
Enter value you want to insert: 213
Enter value you want to insert: 65
Enter value you want to insert: 78
Enter value you want to insert: 23

hashed array is as below
array Index 0 :            65
array Index 1 :            213
array Index 2 :            78
array Index 3 :            234
array Index 4 :            23
array Index 5 :            NULL
array Index 6 :            NULL
array Index 7 :            NULL
array Index 8 :            NULL
array Index 9 :            NULL
```

Searching and hashing