Digit Extraction Hashing.

Fold Boundry hashing

```
#include<iostream>
#include<math.h>
using namespace std;

class Fold_Shift
{
public:
int n,arr[9999],select,i,count,key,temp,flag,k,temp_key,arr_digit,temp_n;

  void get_arr()
  {
   cout<<"Enter the size of array: ";
   cin>>n;

    for(i=0;i<n;i++)
        arr[i]=0;
  count=0;
  }

  void menu()
  {
    while(select!=4)
    {
     cout<<"\n\nChoose from the following:   1.Insert   2.Search   3.Display   4.Exit\n";
     cin>>select;
   switch(select)
   {
      case 1:
         insert();
         break;
      case 2:
         search();
         break;
      case 3:
         display();
         break;
      case 4:
         break;
      default:
         cout<<"\nIncorrect Input!!!";
         break;
   }
  }
}


void insert()
{

  if(count==n)
  {
      cout<<"\nThe array is full!";
```

```
}

else
{
  cout<<"\nEnter the key-value you wish to insert: ";
 cin>>key;

   temp_key=key;

   temp_n=n-1;
   arr_digit=0;
   while(temp_n!=0)
   {
      arr_digit++;
      temp_n/=10;
   }


   int sum=0;
   arr_digit=pow(10,arr_digit);
   while(temp_key!=0)
   {
      sum+=(temp_key%arr_digit);
      temp_key=temp_key/arr_digit;
   }
   sum=sum%arr_digit;
```

```
if(sum>n-1)
{
   sum%=n;
}

if(arr[sum]==0)
{
   arr[sum]=key;
   count++;
}

else
{
   temp=0,flag=0;

   while(arr[sum+temp]!=0)
   {
       if(sum+temp==n-1)
       {
          flag=1;
          break;
       }
    temp++;
```

```
       }

       if(flag==0)
       {
         arr[sum+temp]=key;
        count++;
       }

       else if(flag==1)
       {
          k=0;
          while(arr[k]!=0)
          {
            k++;
          }
        arr[k]=key;
        count++;
       }
    }
  }
}

void search()
{
  cout<<"\nEnter the key-value you wish to search in the array: ";
  cin>>key;
```

```
    temp_key=key;

    temp_n=n-1;
    arr_digit=0;
    while(temp_n!=0)
    {
        arr_digit++;
        temp_n/=10;
    }


    int sum=0;
    arr_digit=pow(10,arr_digit);
    while(temp_key!=0)
    {
        sum+=(temp_key%arr_digit);
        temp_key=temp_key/arr_digit;
    }
    sum=sum%arr_digit;

if(sum>n-1)
{
    sum%=n;
}
```

```
if(arr[sum]==key)
{
   cout<<"\nElement found at position: arr["<<sum<<"]";
}

else
{
   temp=0,flag=0;
   while(arr[sum+temp]!=key)
   {
      if(sum+temp==n-1)
      {
         flag=1;
      }
      temp++;
   }
   if(flag==0)
   {
      cout<<"\nElement found at position: arr["<<sum+temp<<"]";
   }
   else if(flag==1)
   {
      k=0; int verify=0;
      while(arr[k]!=key)
      {
         k++;
```

```
          if(k==n-1 && arr[k]!=key)

          {

            cout<<"\nElement does not exist in the array!";

            verify=1;

            break;

          }

        }

        if(verify==0)

        cout<<"\nElement found at position: arr["<<k<<"]";

      }

    }

  }


  void display()

  {

    cout<<"\nThe elements of the array are: ";

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

    {

      cout<<"\narr["<<i<<"] : "<<arr[i];

    }

  }


};


int main()

{

Fold_Shift obj;
```

```
obj.get_arr();
obj.menu();
return 0;
}
```

Output :

```
Enter the size of array: 5
Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
Enter the key-value you wish to insert: 10

Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
Enter the key-value you wish to insert: 20
Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
Enter the key-value you wish to insert: 30

Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
Enter the key-value you wish to insert: 40
Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
Enter the key-value you wish to insert: 50
Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
1
The array is full!
Choose from the following:   1.Insert   2.Search   3.Display   4.Exit
3
The elements of the array are:
arr[0] : 50
arr[1] : 10
```

```
arr[2] : 20

arr[3] : 30

arr[4] : 40

Choose from the following:    1.Insert    2.Search    3.Display    4.Exit

2

Enter the key-value you wish to search in the array: 10

Element found at position: arr[1]

Choose from the following:    1.Insert    2.Search    3.Display    4.Exit
```