**Program to implement fold shift:**

#include<iostream>

#include<stdio.h>

#include<conio.h>

using namespace std;

class hash

{

int c;

longint array[10];

int count;

public:

hash();

void collision(int,long int);

void print();

void fold(long int);

};

//hash constructor

hash::hash()

{

for(int i=0;i<10;i++)

array[i]=0;

c=0;

}

//print function

void hash::print()

{

cout<<"\n\tAddress \t Key \n";

for(int i=0;i<10;i++)

{

cout<<"\t"<<i<<"\t\t"<<array[i]<<endl;

}

cout<<"\ntotal number of collusion"<<c;

}

//collision

void hash::collision(int add,long int key)

{

if(array[add]==0)

{

array[add]=key;

}

else

{

c++;

collision(add+1,key);

}

}

//fold shift

void hash::fold(long int key)

{

int add;

int sum,sum1,sum2;

int new\_key=key; **123456**

sum=new\_key%1000; 456

new\_key/=1000; 123

sum1=new\_key%1000; 123

new\_key/=1000; 0

sum2=new\_key; 0

add=(sum+sum1+sum2)%10; // (456+123+0)%10=579%10=9

add=add%10;

if(array[add]==0)

{

array[add]=key;

}

else

{

collision(add+1,key);

}

}

//main method

int main()

{

hash s;

longintcount,data;

cout<<"\n enter the no. of key"<<endl;

cin>>count;

cout<<"\nenter the key which is need to be added in the hash array"<<endl;

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

{ cin>>data;

s.fold(data);

}

s.print();

}

**Program to implement fold boundary:**

#include<iostream>

#include<stdio.h>

#include<conio.h>

using namespace std;

class hash

{

int c;

longint array[10];

int count;

public:

hash();

void collision(int,long int);

void print();

voidfold\_boundary(long int);

};

//hash constructor

hash::hash()

{

for(int i=0;i<10;i++)

array[i]=0;

c=0;

}

//print function

void hash::print()

{

cout<<"\n\tAddress \t Key \n";

for(int i=0;i<10;i++)

{

cout<<"\t"<<i<<"\t\t"<<array[i]<<endl;

}

cout<<"\ntotal number of collusion"<<c;

}

//collision

void hash::collision(int add,long int key)

{

if(array[add]==0)

{

array[add]=key;

} else

{

c++;

collision(add+1,key);

}

}

//fold boundary

void hash::fold\_boundary(long int key)

{

int add;

int sum,sum1,sum2,rem,rev=1;

int new\_key=key; //123456

new\_key=new\_key/1000; //123

sum=new\_key%1000; //456

new\_key=key; //123

while(new\_key!=0)

{

rem=new\_key%10; // 3 2 1

rev=rev\*10+rem; //r=30 //32 321

new\_key/=10; //12 1 0

}

new\_key=rev; // 321

sum1=new\_key%1000; // 321

new\_key/=1000000; 0

sum2=new\_key; //0

cout<<sum<<endl;

cout<<sum1<<endl;

cout<<sum2<<endl;

add=(sum+sum1+sum2)%10; //654+321+0=975

cout<<add<<endl;

add=add%10;

cout<<add;

if(array[add]==0)

{

array[add]=key;

}

else

{

collision(add+1,key);

}

}

//main method

int main()

{

hash s;

longintcount,data;

cout<<"\n enter the no. of key"<<endl;

cin>>count;

cout<<"\nenter the key which is need to be added in the hash array"<<endl;

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

{

cin>>data;

s.fold\_boundary(data);

}

s.print();

}