GROUP 4

M8-PAVAN KUMAR M9-SUSHMITA N0-SUPHIYA

N1-LAVANTH N2-SURYA N3-KARTHIK N4-GOPAL RAJU

N5-VENKATESH

AIM:- Write a program to implement different types of hash functions.

DOCUMENTATION:-

Hashing:- Hashing is the process of mapping large amount of data with the help of hashing function.

Hash Table:-

* Data structure used for storing and retrieving data quickly.
* Insertion Based on the key value.

Hash Key:-

* Required data can be searched in hash table by few or more comparisions.

Hash Function:-

* Used to put the data in the Hash table.
* The integer required by hash function is called Hash Key

Hash Key(H)= Key % Table Size

Ex:- If Key=24 and Table size =10 then H= 24%10 = 4

There are 4 methods of Hashing

Mid- Square method:-

* The Key is squared.
* The resultant squared value mid term is used as index.

Ex:- Key = 311

(311)²= 9678321

The hash index is 783, then place 311 into 783 index.

Division method:-

* Depends on remainder of division.

Ex:- 22,24,26,89,75

Table size=10

H(Key)= Key % Table size

H(22)= 22% 10 = 2

H(24)= 24%10 = 4

H(89)= 89%10 = 9

H(75)= 75%10 = 5

Digit Folding:-

* Key is divided into several parts and then combined to produce the Hash Key.

Ex:- 1235678

123,567,8

123+567+8 = 698

* “698” is the index to place the record ‘1235678’

Multiplicative Hash Function:-

* Given record is multiplied with some constant value.

H(Key)= floor(\*P\*(Key\*A))

Algorithm for division hash function:-

Start

Step 1:- Declare the elements

Step 2:- Print No. of elements you want to enter. Read the value of P

Step 3:- Print the size of the table. Read the value of n.

Step 4:- Repeat the following steps untill (i<P)

Initialise i=0

4.1 Print the number

4.2 Read the value of K

4.3 Key= K% n

4.4 assign a[key]=k

Increment the value of i

Step 5:- Repeat the following steps untill (i<100)

Initialise i=0

5.1 if (a[i]==NULL)

Continue

5.2 else

Print the value of index and value of a[i]

5.3 Increment the value of i.

Stop

Program for division hash function:-

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int a[100]={NULL},i,p,key,n,k;

cout<<"number of elements"<<endl;

cin>>p;

cout<<"enter size of table";

cin>>n;

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

{

cout<<"enter the key value"<<endl;

cin>>k;

key=k%n;

a[key]=k;

}

for(i=0;i<100;i++)

{

if(a[i]==NULL)

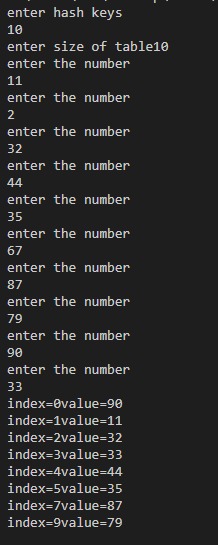
continue;

else

cout<<"index="<<i<<"value="<<a[i]<<endl;

}

}



Algorithm for Multiplication hash function:-

Start

Step 1:- Declare the elements.

Step 2:- Print No.of elements you want to enter in hash functions.

Step 3:- Read the element S.

Step 4:- Print the value of P and read the element.

Step 5:- Repeat the following steps until (i<s)

Initialise i=0

5.1 Print the number

5.2 Read the value of n

5.3 Key= floor(p\*n\*0.619032)

5.4 assign a[key]=n

5.5 increment the value of i.

Step 6:- Repeat the following steps u til!(i<100)

Initialise i=0

5.1 if (a[i]== NULL)

Continue

5.2 else

Print the value of index and value is a[i]

5.3 Increment the value of I

Stop

Program for multiplication hash function:-

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int a[100]={NULL},i,n,s,m,p,key;

cout<<"Enter the how many elements you want to enter in the hash table"<<endl;

cin>>s;

cout<<"Enter the value of p";

cin>>p;

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

{

cout<<"enter the number "<<endl;

cin>>n;

key=floor(p\*n\*0.618032);

a[key]=n;

}

for(i=0;i<100;i++)

{

if(a[i]==NULL)

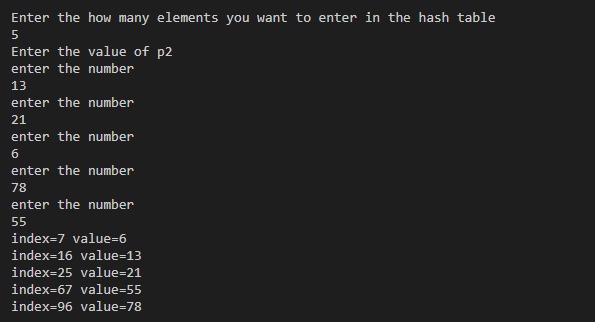
continue;

else

cout<<"index="<<i<<" value="<<a[i]<<endl;

}

}



ALGORTHIM FOR MID SQUARE

START

STEP 1:Declare the elements

STEP 2: Print the no.of elements and read

the value of p.

STEP 3:Repeat the following steps until(i<p)

Initialise i=0

3.1 Read the element k

3.2 initialise key=0 and count=0

3.3 key=k\*k

3.4 z=key

3.5 Repeat the following steps until

(k!=0)

3.5.1 key/=10

3.5.2 count++

3.6 Assign x=z

3.7 q=pow(10,count-1)

3.8 r=x%q

3.9 u=r/10

3.10 a[u]=k

STEP 4: Repeat the following steps until

(i<1000)

Initialise i=0

4.1 if(a[i]==NULL)

Continue

4.2 else

Print the index and value

4.3 increment the value of i(i++)

STOP

PROGRAM-

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int a[1000]={NULL},i,p,key,k,z,count=0,x,r,q,u;

cout<<"enter hash keys"<<endl;

cin>>p;

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

{

cout<<"enter the number"<<endl;

cin>>k;

key=0;

count=0;

key=k\*k;

z=key;

while (key!= 0) {

key/= 10;

count++;

}

x=z;

q=pow(10,count-1);

r=x%q;

u=r/10;

a[u]=k;

}

for(i=0;i<1000;i++)

{

if(a[i]==NULL)

continue;

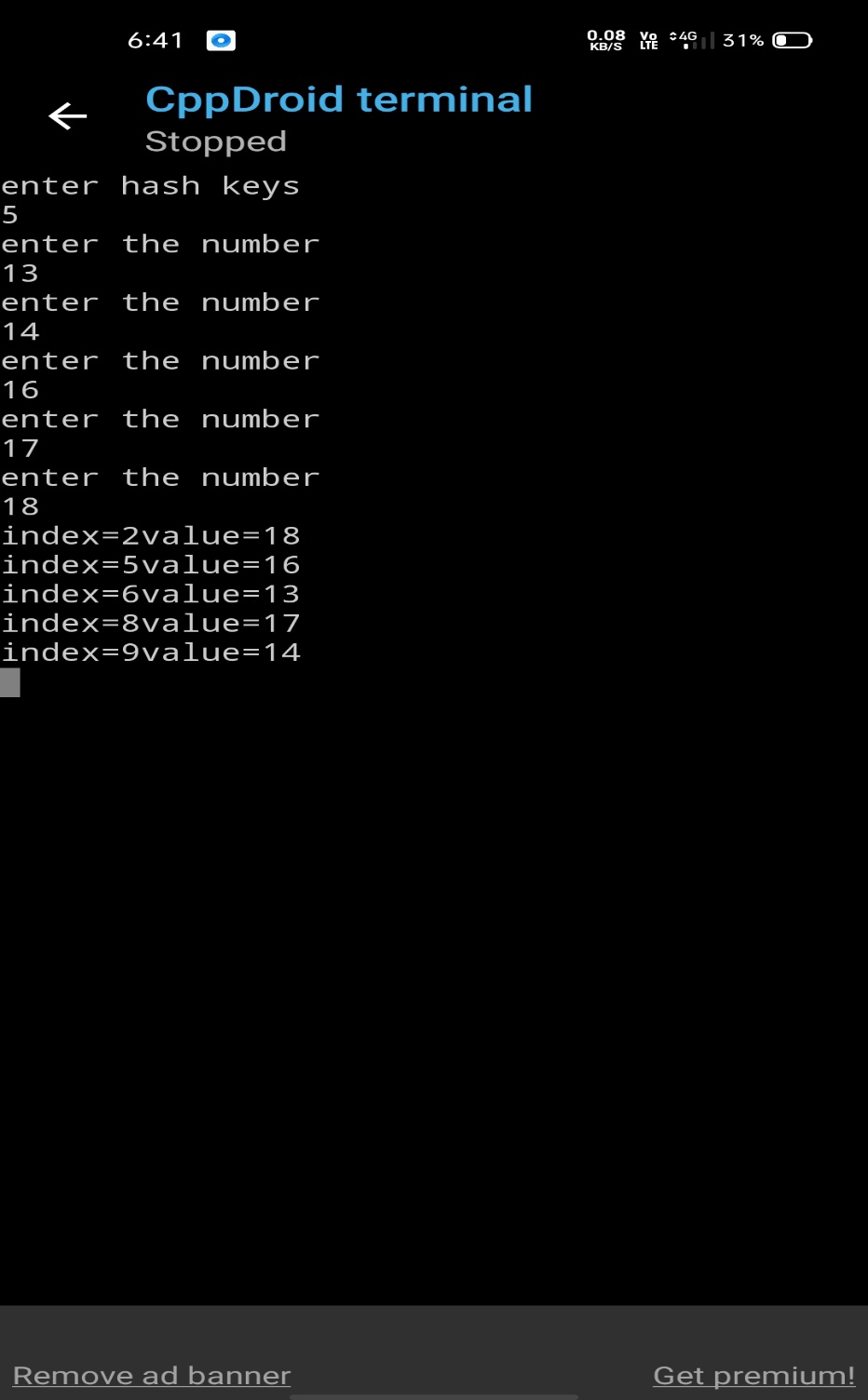
else

cout<<"index="<<i<<"value="<<a[i]<<endl;

}

}

OUTPUT



ALGORTHIM FOR DIGIT FOLDING-

START

STEP 1:Declare the elements

STEP 2: Print the no.of elements and read

the value of p.

STEP 3:Repeat the following steps until(i<p)

Initialise 3.1 Read the element k and sum=0

3.2 initialise temp=0 and sum=0

3.3 Repeat the following steps until (k>100)

Initialise i=0

3.3.1 r=k%100

3.3.2 sum=sum+r

3.3.3 q=k/100

3.3.4 Assign s=sum+k

3.4 Assign s=sum+k

3.5 a[s]=temp

Increment the value of i(i++)

STEP 4: Repeat the following steps until (i<100) Initialise i=0 4.1 if(a[i]==NULL) Continue 4.2 else Print the index and value

4.3 increment the value of i(i++) STOP

PROGRAM FOR DIGIT FOLDING-

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int a[100]={NULL},i,p,key,k,sum=0,q,s,temp,r;

cout<<"enter hash keys"<<endl;

cin>>p;

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

{

cout<<"enter the number"<<endl;

cin>>k;

temp=k;

sum=0;

while(k>100)

{

r=k%100;

sum=sum+r;

q=k/100;

k=q;

}

s=sum+k;

a[s]=temp;

}

for(i=0;i<100;i++)

{

if(a[i]==NULL)

continue;

else

cout<<"index="<<i<<"value="<<a[i]<<endl;

}

}­

OUTPUT-

