|  |
| --- |
| /\* Description : Implementation of hashing |
|  | \*Learner : ARSHEE QURESHI |
|  | \*Created on : 5th October 2017 |
|  | \*/ |
|  |  |
|  | #include<stdio.h> |
|  | #include<math.h> |
|  |  |
|  | #define MAX 10 |
|  |  |
|  |  |
|  | int hash(int k) |
|  | { |
|  | return (k%MAX); |
|  | } |
|  |  |
|  | int linear\_probe(int h) |
|  | { |
|  | return ++h % MAX; |
|  | } |
|  |  |
|  | int quadratic\_probe(int h,int i) |
|  | { |
|  | return (int)(h+pow(i,2))%MAX; |
|  | } |
|  |  |
|  | int double\_hash(int h,int i) |
|  | { |
|  | return (h+i\*linear\_probe(h))%MAX; |
|  | } |
|  |  |
|  | void display(int a[]) |
|  | { |
|  | int i; |
|  | printf("\nElements of array\n"); |
|  | for(i=0;i<=MAX;i++) |
|  | printf("%d\t",a[i]); |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | int a[MAX],i,h,e,count,choice; |
|  | for(i=0;i<=MAX;i++) |
|  | a[i]=-1; |
|  |  |
|  | do |
|  | { |
|  | count=0; |
|  | printf("\n Enter the Number to be stored: "); |
|  | scanf("%d",&e); |
|  | h=hash(e); |
|  | if(a[h]==-1){ |
|  | a[h]=e; |
|  | display(a); |
|  | } |
|  | else |
|  | { |
|  |  |
|  | while(count<MAX){ |
|  | count++; |
|  | printf("\nCollision occured at %d\n",h); |
|  | printf("\n Pease enter your choice for resolution\n"); |
|  | printf("\n1.LINEAR PROBING\n2.QUADRATIC PROBING\n3.DOUBLE HASHING\n"); |
|  | scanf("%d",&choice); |
|  |  |
|  | switch(choice){ |
|  | case 1: |
|  | h=linear\_probe(h); |
|  | break; |
|  | case 2: |
|  | h=quadratic\_probe(h,count); |
|  | break; |
|  | case 3: |
|  | h=double\_hash(h,count); |
|  | break; |
|  | default : |
|  | printf("\n Invalid choice\n"); |
|  | count--; |
|  | break; |
|  | } |
|  |  |
|  | if(a[h]==-1) |
|  | { |
|  | a[h]=e; |
|  | display(a); |
|  | break; |
|  | } |
|  | } |
|  | } |
|  | if(count==MAX) |
|  | { |
|  | printf("\n Array is FULL\nor\nCollision Resolution not successfull"); |
|  | return 0; |
|  | } |
|  | }while(1); |
|  | } |
|  |  |
|  |  |
|  | /\*Enter the Number to be stored: 15 |
|  |  |
|  | Elements of array |
|  | -1 -1 -1 -1 -1 15 -1 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 13 |
|  |  |
|  | Elements of array |
|  | -1 -1 -1 13 -1 15 -1 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 12 |
|  |  |
|  | Elements of array |
|  | -1 -1 12 13 -1 15 -1 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 4 |
|  |  |
|  | Elements of array |
|  | -1 -1 12 13 4 15 -1 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 6 |
|  |  |
|  | Elements of array |
|  | -1 -1 12 13 4 15 6 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 1 |
|  |  |
|  | Elements of array |
|  | -1 1 12 13 4 15 6 -1 -1 -1 -1 |
|  | Enter the Number to be stored: 69 |
|  |  |
|  | Elements of array |
|  | -1 1 12 13 4 15 6 -1 -1 69 -1 |
|  | Enter the Number to be stored: 36 |
|  |  |
|  | Collision occured at 6 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 1 |
|  |  |
|  | Elements of array |
|  | -1 1 12 13 4 15 6 36 -1 69 -1 |
|  | Enter the Number to be stored: 48 |
|  |  |
|  | Elements of array |
|  | -1 1 12 13 4 15 6 36 48 69 -1 |
|  | Enter the Number to be stored: 52 |
|  |  |
|  | Collision occured at 2 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 3 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 7 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 6 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 2 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 7 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 3 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 2 |
|  |  |
|  | Collision occured at 2 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 3 |
|  |  |
|  | Collision occured at 6 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 3 |
|  |  |
|  | Collision occured at 9 |
|  |  |
|  | Pease enter your choice for resolution |
|  |  |
|  | 1.LINEAR PROBING |
|  | 2.QUADRATIC PROBING |
|  | 3.DOUBLE HASHING |
|  | 1 |
|  |  |
|  | Elements of array |
|  | 52 1 12 13 4 15 6 36 48 69 -1 |
|  | Array is FULL |
|  | or |
|  | Collision Resolution not successfull |
|  |  |
|  | ------------------ |
|  | (program exited with code: 0) |
|  | Press return to continue |
|  |  |
|  | \*/ |