Experiment 18
Date 22-12-2022

HASHING

Aim

Write a C program to implement a Hash table using (Let the size of the hash table be 10 so that the index varies from 0 to 9)
a. Chaining method.
b. Linear Probing for collision resolution.

1 Sorting Algorithms

1.1 Algorithm

```
Algorithm chaining method

Step 1:Start
Step 2:Input the number of key values.
Step 3:An array of linked list is created with data and link and initialize
       each list with  NULL value.
Step 4:Each value is entered and the hashindex is found.The value is stored.
Step 5:If the index value is repeating then the end of the link is found and
        inserted at end.
Step 6:Display the array.
Step 7:Stop

Algorithm linear probing
Step 1:Start
Step 2:Create an array of size 10.
Step 3:Enter the key values.
Step 4:Find the index for each key value using hash function.
Step 5:If collision of indices occur,the next free index is found and store the
        c key value.If index exceeds the size of array then searching starts from
        index 0.
Step 6:Print the array
Step 7:Stop
```

1.2 Program

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{
int data;
struct node*link;
} node;
int size=10;
node* a[30];
node* new;
node* temp;
node* ptr;
int b[10];
void insert1(int val);
void display();
void insert2(int val);
void dis();
void insert1(int val)
{
new=(node*)malloc(sizeof(node));
new->data=val;
new->link=NULL;
int k=val%size;
if(a[k]==NULL)
{
a[k]=new;
}
else
{
temp=a[k];
while(temp->link!=NULL)
{
temp=temp->link;
}
temp->link=new;
}
}
void display()
{
int i;
for(i=0;i<size;i++)
```

```
{
ptr=a[i];
printf("%d :",i);
while(ptr!=NULL)
{
printf("%d\t",ptr->data);
ptr=ptr->link;
}
printf("\n");
}
}
void insert2(int val)
{
int k=val%size;
if(b[k]==-1)
{
b[k]=val;
}
else
{
int i;
int f=0;
for(i=k+1;i<size;i++)
{
if(b[i]==-1)
{
b[i]=val;
f=1;
break;
}
}
if(f==0)
{
for(i=0;i<k;i++)
{
if(b[i]==-1)
{
b[i]=val;
break;
}
}
}
}
}
void dis()
{
```

```
int i;
for(i=0;i<size;i++)
{
if(b[i]==-1)
{
printf("%d :\n",i);
}
else
{
printf("%d :%d\n",i,b[i]);
}
}
}
void main()
{
int i,n,x,choice;
for(i=0;i<size;i++)
{
b[i]=-1;
}
printf("Enter the number of inputs:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter the input:");
scanf("%d",&x);
insert1(x);
insert2(x);
}
printf("\nChaining method\n");
display();
printf("\nLinear Probing\n");
dis();
}
```

1.3 Sample Output

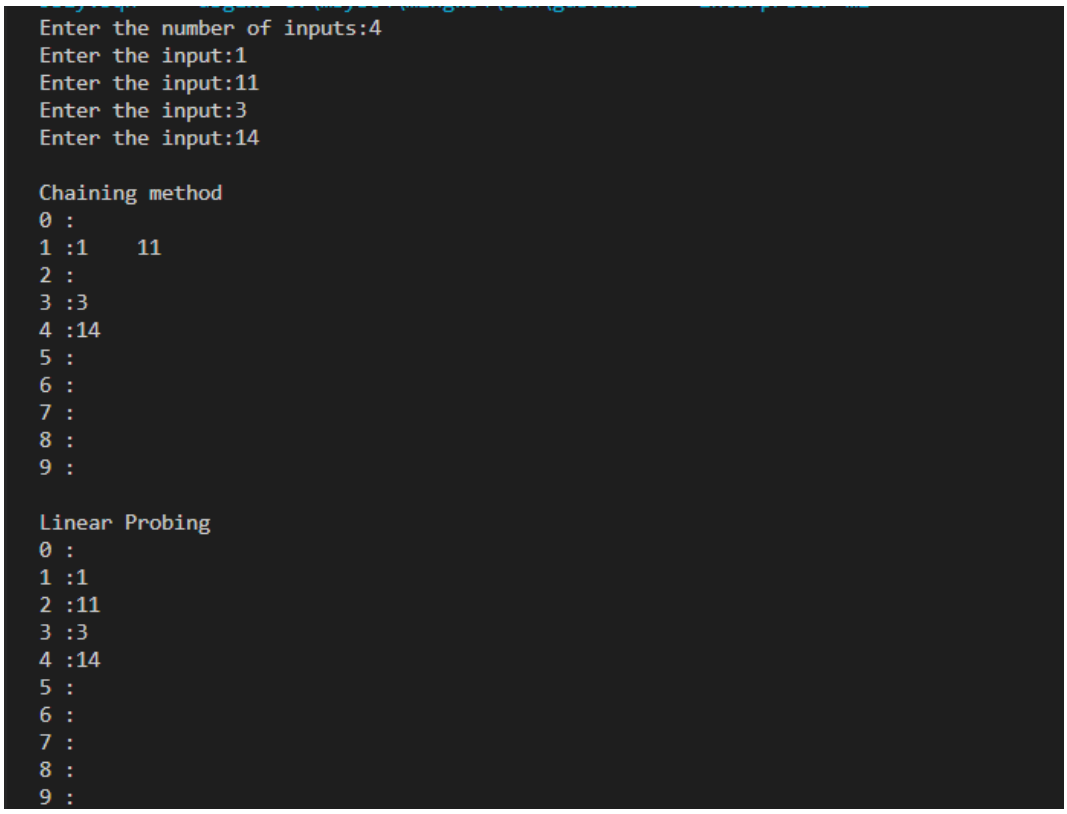

Figure 1: Input and Output

1.4 Result

Implemented a hash table using chaining method and linear probing for collision resolution by using separate functions for each.