***PROGRAM:***

#include <stdio.h>

#include<conio.h> // C program for the above approach

#include <stdio.h>

#include <stdlib.h>

struct HashNode

{

int key;

int value;

};

const int capacity = 20;

int size = 0;

struct HashNode\*\* arr;

struct HashNode\* dummy;

// Function to add key value pair

void insert(int key, int v)

{

struct HashNode\* temp=(struct HashNode\*)malloc(sizeof(struct HashNode));

temp->key=key;

temp->value=v;

// Apply hash function to find

// index for given key

int hashIndex=key%capacity;

// Find next free space

while (arr[hashIndex] != NULL

&&arr[hashIndex]->key != key

&&arr[hashIndex]->key != -1) {

hashIndex++;

hashIndex %= capacity;

}

// If new node to be inserted

// increase the current size

if (arr[hashIndex] == NULL || arr[hashIndex]->key == -1)

size++;

arr[hashIndex]=temp;

}

// Function to delete a key value pair

int delete(int key)

{

// Apply hash function to find

// index for given key

int hashIndex = key % capacity;

// Finding the node with given

// key

while (arr[hashIndex] != NULL) {

// if node found

if (arr[hashIndex]->key == key) {

// Insert dummy node here

// for further use

arr[hashIndex] = dummy;

// Reduce size

size--;

// Return the value of the key

return 1;

}

hashIndex++;

hashIndex%=capacity;

}

// If not found return null

return 0;

}

// Function to search the value

// for a given key

int find(int key)

{

// Apply hash function to find

// index for given key

int hashIndex=(key%capacity);

int counter=0;

// Find the node with given key

while(arr[hashIndex]!=NULL){

int counter=0;

// If counter is greater than

// capacity

if (counter++ > capacity)

break;

// If node found return its

// value

if (arr[hashIndex]->key == key)

return arr[hashIndex]->value;

hashIndex++;

hashIndex%=capacity;

}

// If not found return

// -1

return -1;

}

//Driver code

int main()

{

// Space allocation

arr = (struct HashNode\*\*)malloc(sizeof(struct HashNode\*)\*capacity);

// Assign NULL initially

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

arr[i] = NULL;

dummy=(struct HashNode\*)malloc(sizeof(struct HashNode));

dummy->key = -1;

dummy->value = -1;

insert(1, 5);

insert(2, 15);

insert(3, 20);

insert(4, 7);

if (find(4) != -1)

printf("Value of Key 4 = %d\n", find(4));

else

printf("Key 4 does not exists\n");

if(delete(4))

printf("Node value of key 4 is deleted ""successfully\n");

else{

printf("Key does not exists\n");

}

if (find(4) != -1)

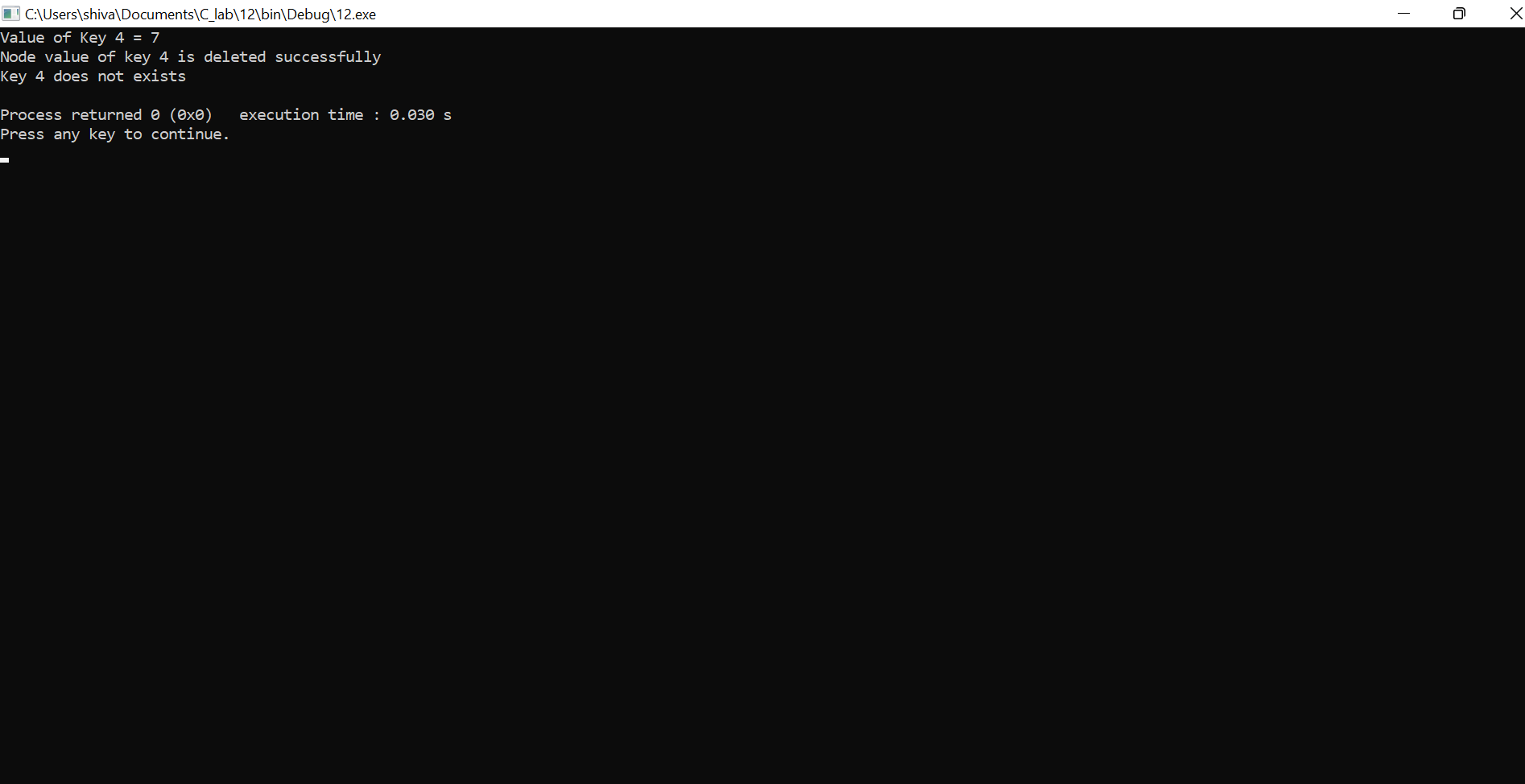
printf("Value of Key 4 = %d\n", find(4));

else

printf("Key 4 does not exists\n");

}

***OUTPUT:***

******