PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH **Year:** 2024 – 25 **Semester:** I

Name: Sonawane Prachi Mahendra **PRN:** 124B2B018

Department : Computer Engineering **Division:** B

Course : Data Structures Laboratory **Course Code:** BCE23PC02

Date: 18/10/24

Assignment – 11

- **Aim:**

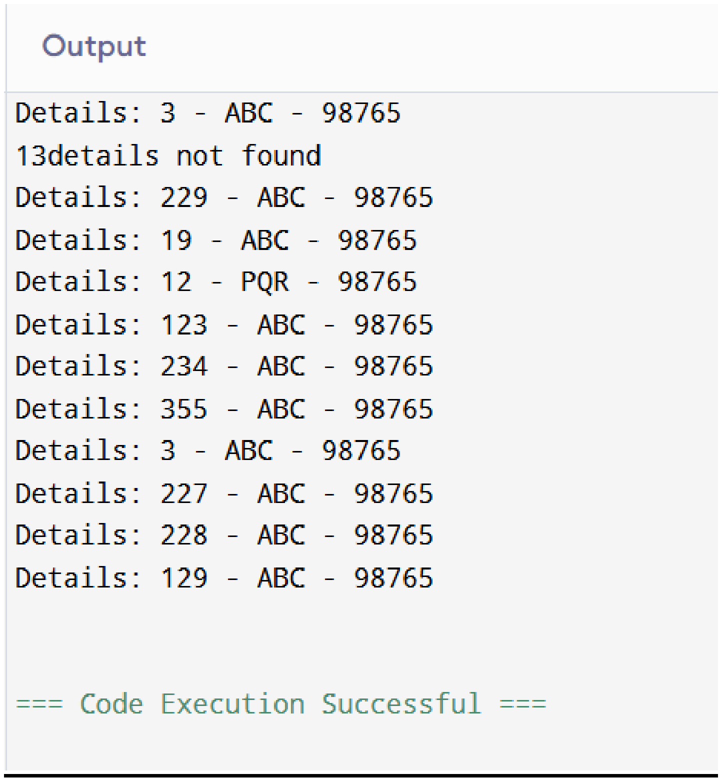
  Consider an employee database of N employees. Make use of a hash table implementation to quickly look up the employer's id number.

- **Source Code :**

```
#include<iostream>
using namespace std;
class Employee{

        int EmpID;
        string Name;
        int contact;

        public:
        int index;
        Employee(){
                EmpID = -1;
                Name = "";
                contact = -1;
                index = -1;
        }
        void setID(int id){EmpID = id;}
        void setName(int n){Name = n;}
        void setContact(int c){contact = c;}
        void setIndex(int i){ index = i;}

        int getID(){ return EmpID;}
        void setEmployee(int EmpID,string N,int contact,int index){
                this->EmpID = EmpID;
                Name = N;
```

```
            this->contact = contact;
            this->index = index;
        }
        void printEmployee(){
            cout<<"Details: "<<EmpID<<" - "<<Name<<" - "<<contact<<endl;
        }

};
class HashTable{
        int tableSize;
        Employee *ht;
        public:
        HashTable(int size){
            tableSize = size;
            ht = new Employee[tableSize];
        }
        int hash(int value){
            return(value%tableSize);
        }
        void insertIntoHT(int EmpID,string N,int contact){
            int ToBeInsertedAt = hash(EmpID);
            if(ht[ToBeInsertedAt].index == -1){ //position is empty, can insert here
                ht[ToBeInsertedAt].setEmployee(EmpID,N,contact,ToBeInsertedAt);
            }
            else{
                for(int i=0;i<tableSize;i++){
                    ToBeInsertedAt = (ToBeInsertedAt+1)%tableSize;

                    if(ht[ToBeInsertedAt].index == -1){

        ht[ToBeInsertedAt].setEmployee(EmpID,N,contact,ToBeInsertedAt);
                        return;
                    }
                }
                cout<<"HashTable is full"<<endl;
            }
        }
        void searchInHT(int EmpID){
            int ToBeInsertedAt = hash(EmpID);
            if(ht[ToBeInsertedAt].getID() == EmpID){ //data found
                ht[ToBeInsertedAt].printEmployee();
            }
            else{
                for(int i=0;i<tableSize;i++){
                    ToBeInsertedAt = (ToBeInsertedAt+1)%tableSize;

                    if(ht[ToBeInsertedAt].getID() == EmpID){
                        ht[ToBeInsertedAt].printEmployee();
                        return;
                    }
                }
                cout<<EmpID<<"details not found"<<endl;
            }
        }
        void displayHT(){
            for(int i=0;i<tableSize;i++)
                ht[i].printEmployee();
        }

};

int main(){
        HashTable ht1(10);
        ht1.insertIntoHT(123,"ABC",98765);
        ht1.insertIntoHT(12,"PQR",98765);
        ht1.insertIntoHT(355,"ABC",98765);
        ht1.insertIntoHT(234,"ABC",98765);
        ht1.insertIntoHT(129,"ABC",98765);
        ht1.insertIntoHT(3,"ABC",98765);
        ht1.insertIntoHT(229,"ABC",98765);
        ht1.insertIntoHT(227,"ABC",98765);
        ht1.insertIntoHT(228,"ABC",98765);
        ht1.insertIntoHT(19,"ABC",98765);
        ht1.searchInHT(3);
        ht1.searchInHT(13);
        ht1.displayHT();
        /*ht1.deleteEmployee(100);
        ht1.deleteEmployee(129);
        ht1.deleteEmployee(229);

        ht1.displayHT();*/

}
```

- **<u>Screen Shot of Output :</u>**



Output

```
Details: 3 - ABC - 98765
13details not found
Details: 229 - ABC - 98765
Details: 19 - ABC - 98765
Details: 12 - PQR - 98765
Details: 123 - ABC - 98765
Details: 234 - ABC - 98765
Details: 355 - ABC - 98765
Details: 3 - ABC - 98765
Details: 227 - ABC - 98765
Details: 228 - ABC - 98765
Details: 129 - ABC - 98765


=== Code Execution Successful ===
```

- **Conclusion:**
  Hence, we studied about hashing with its program