ASSIGNMENT NO 4

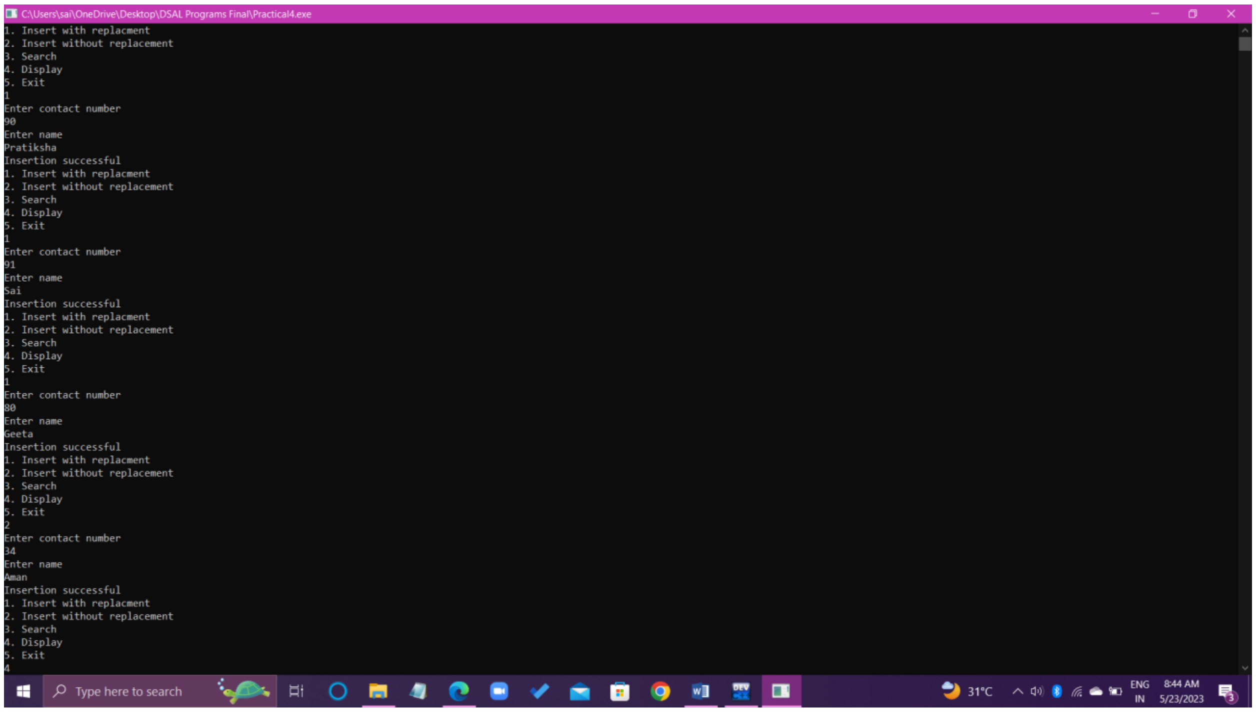
Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. Make use of two collision handling techniques and compare them using number of comparisons required to find a set of telephone numbers (Note: Use linear probing with replacement and without replacement

```
#include <iostream>
using namespace std;

class Pair{
long data;
string value;

public:

Pair(){
data = -1;
value = "null";
}

Pair(long no, string n){
data = no;
value = n;
}

friend class HashTable;
};

class HashTable{
int size = 10;
Pair *arr[10];

int hash(long key){
return key%10;
}


void place_at_correct_position(Pair *n, long key){
int index = hash(key);

for(int i = 0; i < 10; i++){
int curr_index = (index + i)%size;

if(arr[curr_index]->data == -1){
arr[curr_index] = n;
```

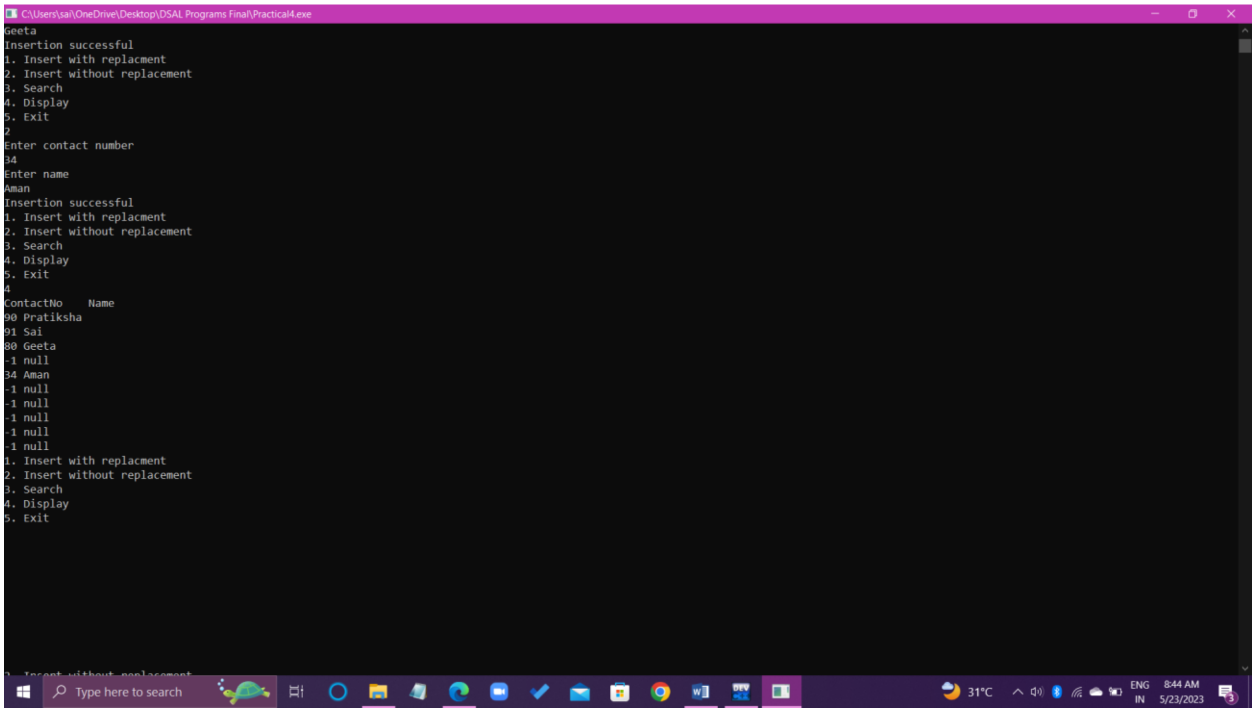
```
break;
}
}
}

public:

HashTable(){
for(int i = 0; i < 10; i++){
arr[i] = new Pair();
}
}

void display(){
cout<<"ContactNo    Name"<<endl;
for(int i = 0; i < size; i++){

cout<<arr[i]->data<<" "<<arr[i]->value<<endl;
}
}

bool isPresent(long key){
for(int i = 0; i < size; i++){
if(arr[i]->data == key){
return true;
}else{
return false;
}
}
}

void without_replacement(long key, string value){
Pair *p = new Pair(key, value);

int index = hash(key);

for(int i = 0; i < 10; i++){
int curr_index = (index + i)%size;

if(!isPresent(key)){
if(arr[curr_index]->data == -1){
arr[curr_index] = p;
break;
}
}else{
```

```
cout<<"Key already exists"<<endl;
break;
}
}
}

void with_replacement(long key, string value){
Pair *p = new Pair(key, value);

int index = hash(key);

if(arr[index]->data == -1){
arr[index] = p;
}

else if(hash(arr[index]->data) != key%size){
Pair *q = new Pair;
q = arr[index];
arr[index] = p;
place_at_correct_position(q, q->data);
}
else{
place_at_correct_position(p, key);
}

}

int search(long key){
int count = 0;
int index = hash(key);

for(int i = 0; i < 10; i++){
int curr_index = (index + i)%size;
count++;

if(arr[curr_index]->data == key){
cout<<"Found after "<<count<<" number of comparison"<<endl;
cout<<"key "<<arr[curr_index]->data<<" value "<<arr[curr_index]->value<<endl;
return count;
}
}

cout<<"Key is not present"<<endl;
}
```

```cpp
};

int main() {

HashTable h1;
int inp = 0;
long key = 0;
bool flag = true;
string name = "";

do{
cout<<"1. Insert with replacment"<<endl
<<"2. Insert without replacement"<<endl
<<"3. Search"<<endl
<<"4. Display"<<endl
<<"5. Exit"<<endl;
cin>>inp;

switch(inp){
case 1:
cout<<"Enter contact number "<<endl;
cin>>key;
cout<<"Enter name "<<endl;
cin>>name;

h1.with_replacement(key, name);
cout<<"Insertion successful"<<endl;
break;

case 2:
cout<<"Enter contact number "<<endl;
cin>>key;
cout<<"Enter name "<<endl;
cin>>name;

h1.without_replacement(key, name);
cout<<"Insertion successful"<<endl;
break;

case 3:
cout<<"Enter what you want to search "<<endl;
cin>>key;

h1.search(key);
break;
case 4:
h1.display();
break;

case 5:
flag = false;
cout<<"Exiting..";
break;

default:
cout<<"Choose correct option"<<endl;
break;
}


}while(flag);

return 0;
}
```


C:\Users\sai\OneDrive\Desktop\DSAL Programs Final\Practical4.exe
Type here to search
31°C
ENG
IN
8:44 AM
5/23/2023

```
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
1
Enter contact number
90
Enter name
Pratiksha
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
1
Enter contact number
91
Enter name
Sai
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
1
Enter contact number
80
Enter name
Geeta
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
2
Enter contact number
34
Enter name
Aman
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
4
```



C:\Users\sai\OneDrive\Desktop\DSAL Programs Final\Practical4.exe

```
Geeta
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
2
Enter contact number
34
Enter name
Aman
Insertion successful
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
4
ContactNo    Name
90 Pratiksha
91 Sai
80 Geeta
-1 null
34 Aman
-1 null
-1 null
-1 null
-1 null
-1 null
1. Insert with replacment
2. Insert without replacement
3. Search
4. Display
5. Exit
```