***Task 1***

#include<iostream>

#include<string>

using namespace std;

struct Nodes

{

string Data;

Nodes\* Next;

int Key;

bool Flag = true;

};

class Library

{

public:

Nodes\* Book;

int Num;

Library()

{

Book = NULL;

Num = 0;

}

void Insertion()

{

int S;

int B\_Number;

string B\_Name;

cout << "Please Enter Total Number of Books : ";

cin >> Num;

Book = new Nodes[Num];

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

{

cout << endl << endl;

cout << "Please Enter The Data Of Book : " << i+1 << endl;

cout << "Book Name: ";

cin >> B\_Name;

cout << "Book Number: ";

cin >> B\_Number;

S = Hashing(B\_Name, Num);

if (Book[S].Flag)

{

Book[S].Data = B\_Name;

Book[S].Key = B\_Number;

Book[S].Flag = false;

Book[S].Next = NULL;

}

else

{

Nodes\* Foo = &(Book[S]);

while (Foo->Next != NULL)

{

Foo = Foo->Next;

}

Foo->Next = new Nodes;

Foo->Next->Data = B\_Name;

Foo->Next->Key = B\_Number;

Foo->Next->Flag = false;

Foo->Next->Next = NULL;

}

}

}

int Hashing(string N, int n)

{

int S, Sum = 0;

S = N.length();

char temp[1000 + 1];

for (int i = 0; i < 1000; i++)

{

Sum = Sum + temp[i];

}

return (Sum % n);

}

void Searching(string N)

{

int search;

search = Hashing(N, Num);

if (Book[search].Next == NULL)

{

if (Book[search].Data == N)

{

cout << "Book Found Successfully...! " << endl;

cout << "The Number Of Book Is: " << Book[search].Key << endl;

}

else

{

cout << "Book Name doesn't Found" << endl;

}

}

else

{

Nodes\* Foo = &(Book[search]);

while (Foo != NULL)

{

if (Foo->Data == N)

{

cout << "Book Found Successfully...! " << endl;

cout << "The Number Of Book Is: " << Foo->Key << endl;

break;

}

else

{

Foo = Foo->Next;

}

}

if (Foo == NULL)

{

cout << "Book Not Found...!" << endl;

}

}

}

};

int main()

{

string Name;

int n;

Library L1;

L1.Insertion();

cout << endl << endl;

cout << "Please Enter Total Number Of Books U Want To Search: ";

cin >> n;

cout << endl << endl;

cout << "Enter Searching Book Name : ";

for (int i = 1; i <= n; i++)

{

cout << "Please Enter " << i <<" Book Name : ";

cin >> Name;

L1.Searching(Name);

cout << endl;

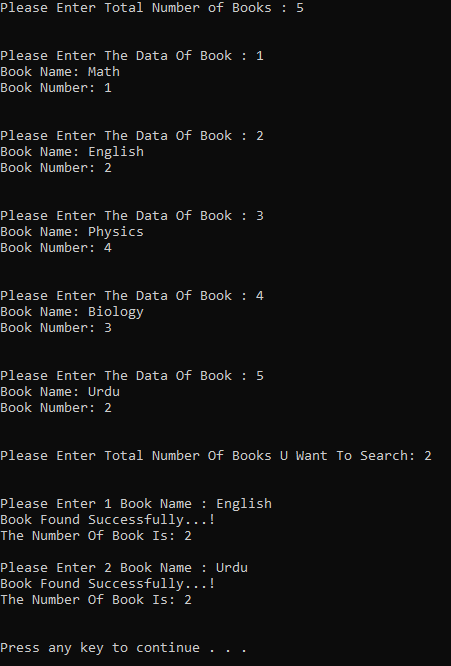
}

cout << endl;

system("pause");

return 0;

}



***TASK 2***

#include<iostream>

#include<string>

using namespace std;

struct node

{

string name;

int val;

node\* next = NULL;

};

class Library

{

public:

int Num;

bool\* Visited;

string\* Array;

int probing;

Library::Library()

{

Array = new string[Num];

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

{

Array[i] = "";

}

Visited = new bool[Num];

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

{

Visited[i] = false;

}

}

void Insertion(string val, int val1)

{

int M = val[0] % Num;

int key = 0;

int i = 1;

key = M;

if (Visited[M] == false)

{

Array[M] = val;

Visited[M] = true;

}

else

{

while (Visited[key] == true)

{

key = (M + i \* i) % Num;

i++;

}

Array[key] = val;

Visited[key] = true;

}

}

void Display()

{

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

{

cout << Array[i] << " ";

}

cout << endl;

}

void Searching(string val)

{

int M = val[0] % Num;

int key = 0;

int i = 1;

key = M;

if (Array[key] == val)

{

cout << "Book Found Successfully...!" << endl;

}

else

{

while (Visited[key] == true && Array[key] != val)

{

key = (M + i \* i) % Num;

i++;

}

if (Array[key] == val)

{

cout << "Book Found Successfully...!" << endl;

}

else

{

cout << "Book Not Found...!" << endl;

}

}

}

};

int main()

{

Library L1;

int Num = 0;

string Name;

cout << "Please Enter The Total Number Of Books: ";

cin >> L1.Num;

L1.Array = new string[L1.Num];

cin.ignore();

for (int i = 0; i < L1.Num; i++)

{

cout << endl << endl;

cout << "Please Enter The Data Of Book: " << i + 1 << endl;

cout << "Book Name: ";

getline(cin, Name);

cout << "Book Number: ";

cin >> Num;

cin.ignore();

L1.Insertion(Name, Num);

}

cout << endl << endl;

cout << "Please Enter Book Name U Want To Serach:" << endl;

cin >> Name;

L1.Searching(Name);

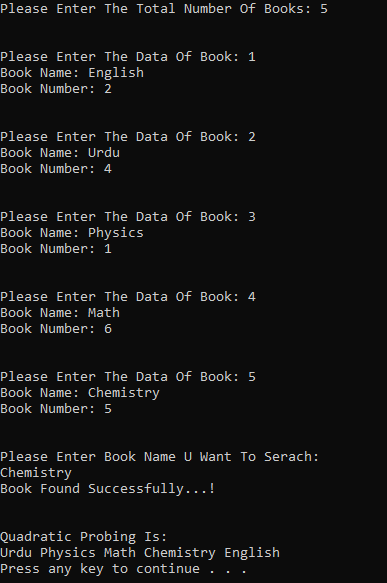
cout << endl << endl;

cout << "Quadratic Probing Is: " << endl;

L1.Display();

system("pause");

}



***TASK 3:***

#include<iostream>

#include<string>

using namespace std;

struct Nodes

{

string Number;

string Name;

string loc;

char Index;

Nodes\* Next = NULL;

Nodes()

{

Number = "";

Name = "";

}

};

class Library

{

public:

Nodes\* Head;

Nodes\* Tail;

int n;

Library::Library()

{

Head = NULL;

Tail = NULL;

}

void Insertion(Library\*& Array, string number, string name, string location)

{

int M = name[0] % n;

if (M < 0)

{

M = M \* -1;

}

if (M < n)

{

Nodes\* Foo = Array[M].Head;

while (Foo != NULL)

{

if (Foo->Index == -1)

{

Foo->Index = name[0];

Foo->Name = name;

Foo->Number = number;

Foo->loc = location;

return;

}

Foo = Foo->Next;

}

Array[M].List(number, name[0], name, location);

}

}

void List(string Num, char Index, string N, string L)

{

Nodes\* newNodes = new Nodes;

newNodes->Number = Num;

newNodes->Name = N;

newNodes->loc = L;

newNodes->Index = Index;

if (Head == NULL)

{

Head = newNodes;

Tail = newNodes;

return;

}

else

{

Tail->Next = newNodes;

Tail = newNodes;

Tail->Next = NULL;

}

}

void Library::Searching(Library\* Array, string name)

{

int M = name[0] % n;

Nodes\* Foo = Array[M].Head;

while (Foo != NULL)

{

if (Foo->Name == name)

{

cout << "Book Food Successfully...!" << endl;

return;

}

Foo = Foo->Next;

}

cout << "Book Not found...!" << endl;

}

void Display(Library\* Array)

{

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

{

Nodes\* Foo = Array[i].Head;

cout << i << " ";

while (Foo != NULL)

{

cout << Foo->Name << "->";

Foo = Foo->Next;

}

cout << endl;

}

}

};

int main()

{

Library\* L1 = 0, L2;

string Name, Location, Number;

cout << "Please Enter The Total NUmber Of Books: ";

cin >> L2.n;

cin.ignore();

L1 = new Library[L2.n];

for (int i = 0; i < L2.n; i++)

{

cout << endl << endl;

cout << "Please Enter The Data Of Book: " << i + 1 << endl;

cout << "Book Name: ";

getline(cin, Name);

cout << "Book Number: ";

getline(cin, Number);

L2.Insertion(L1, Number, Name, Location);

}

cout << endl << endl;

cout << "Please Enter The Element U Want To Serach: " << endl;

getline(cin, Name);

L2.Searching(L1, Name);

cout << endl << endl;

cout << "Seperate Chaining: " << endl;

L2.Display(L1);

cout << endl;

system("pause");

}

