Department of IT and Computer Science

Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan

COMP-201L Data Structures and Algorithms Lab

Lab Report 09

Class: Computer Science

Name: Yaseen Ejaz Ahmed

Registration No.: B20F0283CS014

Semester: Third

Submitted to: Engr. Rafi Ullah

_____

Instructor Signature

Lab No. 9

Hashing

Objectives:

To understand:

- Hash Function
- Hash Table
- Collision Resolution Techniques
- Separate Chaining
- Linear Chaining
- Quadratic Probing
- Double Hashing
- Applications

Tools/Software Required:

C++ Compiler

Introduction:

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.
Some examples of how hashing is used in our lives include:

- In universities, each student is assigned a unique roll number that can be used to retrieve information about them.
- In libraries, each book is assigned a unique number that can be used to determine information about the book, such as its exact position in the library or the users it has been issued to etc.

In both these examples the students and books were hashed to a unique number.

Lab Tasks:

Closed Hashing

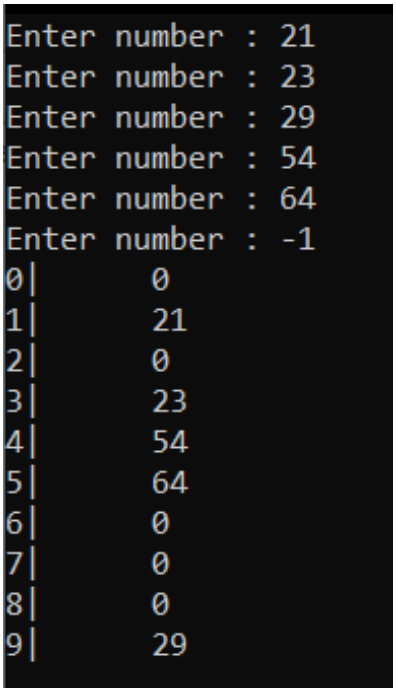
Lab Task 01: Linear Hashing

Code:

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

int main()
{
	int arr[10];

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

	int a,b;
	do
	{
		cout<<"Enter number : ";
		cin>>a;

		b=a%10;

		if(a!=-1)
		{
			while(arr[b]!=0)
			{
				arr[++b];

				if(b==10)
				b=0;
			}

			arr[b]=a;
```

```
            }
        }
        while(a!=-1);

        for(int i=0;i<10;i++)
        {
            cout<<i<<"|";
            cout<<"\t"<<arr[i]<<endl;
        }
}
```

Output:



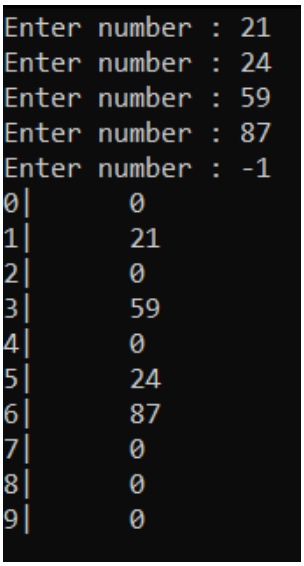
Lab Task 02: Quadratic Hashing

Code:

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

int main()
{
    int arr[10];

    for(int i=0;i<10;i++)
    {
        arr[i]=0;
```

```
	}

	int a,b,i=0;
	do
	{
		cout<<"Enter number : ";
		cin>>a;

		b=a%10;

		b=(b+i*i)%10;

		if(a!=-1)
		{
			while(arr[b]!=0)
			{
				arr[++b];

				if(b==10)
				b=0;
			}

			arr[b]=a;
		}
		i++;
	}
	while(a!=-1);

	for(int i=0;i<10;i++)
	{
		cout<<i<<"|";
		cout<<"\t"<<arr[i]<<endl;
	}
}
```

Output:



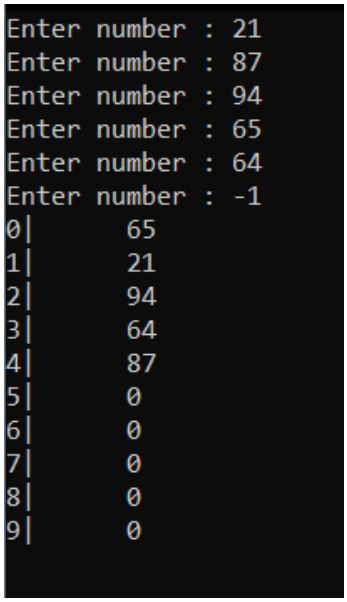
Lab Task 03: Double Hashing

Code:

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

int main()
{
	int arr[10];

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

	int a,b,i=0;
	do
	{
		cout<<"Enter number : ";
		cin>>a;

		b=a%10;

		b=(b+i*a)%10;
```

```
            if(a!=-1)
            {
                while(arr[b]!=0)
                {
                    arr[++b];

                    if(b==10)
                    b=0;
                }

                arr[b]=a;
            }
            i++;
        }
        while(a!=-1);

        for(int i=0;i<10;i++)
        {
            cout<<i<<"|";
            cout<<"\t"<<arr[i]<<endl;
        }
}
```

Output:



Open Hashing

Lab Task 04: Implement hashing using linked list

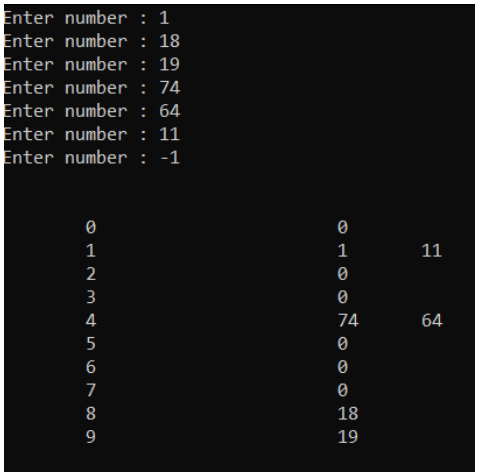
Code:

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

class node
{
	private:
		node* head=NULL;
		node* next;
		node* next1;
		node* ptr;
		node*ptr1;
		int data;

	public:

	void start()
	{
		node* temp=new node;
		temp->data=0;
		temp->next=NULL;
		temp->next1=NULL;

		head=temp;
		ptr=head;
		for(int i=1;i<=10;i++)
		{
			node* temp1=new node;
		   temp1->data=0;
		   temp1->next=NULL;
		   temp1->next1=NULL;

		   ptr->next=temp1;
		   ptr=temp1;
		}
```

```
}

void Push(int value,int count)
{
        ptr=head;

        for(int i=1;i<=count;i++)
        {
                ptr=ptr->next;
        }

        if(ptr->data==0)
        ptr->data=value;

        else
        {
                node* temp=new node;
                temp->data=value;
                temp->next=NULL;
           temp->next1=NULL;

           ptr->next1=temp;
        }
}

void showAll()
{
        cout<<endl<<endl;
        ptr=head;
        int i=0;
        while(ptr->next!=NULL)
        {
                cout<<"\t"<<i++<<"\t\t\t"<<ptr->data;

                ptr1=ptr;

                if(ptr1->next1!=NULL)
                {
                        ptr1=ptr1->next1;
                        while(ptr1->next1!=NULL)
                {
                        cout<<"\t"<<ptr1->data;
```

```
                    ptr1=ptr1->next1;
                }
                cout<<"\t"<<ptr1->data;
                }
                cout<<endl;
                ptr=ptr->next;
            }
        }
};

int main()
{
        node h;
        h. start ();
        int a,b;
        do
        {
            cout<<"Enter number : ";
            cin>>a;

            b=a%10;

            if(a!=-1)
            h.Push(a,b);
        }
        while(a!=-1);
        h.showAll();
}
```

Output:



Lab Task 5: Implement all types of Hashing into one program

Code:

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

class node
{
	private:
		node* head=NULL;
		node* next;
		node* next1;
		node* ptr;
		node*ptr1;
		int data;

	public:

	void start()
	{
		node* temp=new node;
		temp->data=0;
		temp->next=NULL;
		temp->next1=NULL;

		head=temp;
		ptr=head;
		for(int i=1;i<=10;i++)
		{
			node* temp1=new node;
		   temp1->data=0;
		   temp1->next=NULL;
		   temp1->next1=NULL;

		   ptr->next=temp1;
		   ptr=temp1;
		}
	}
```

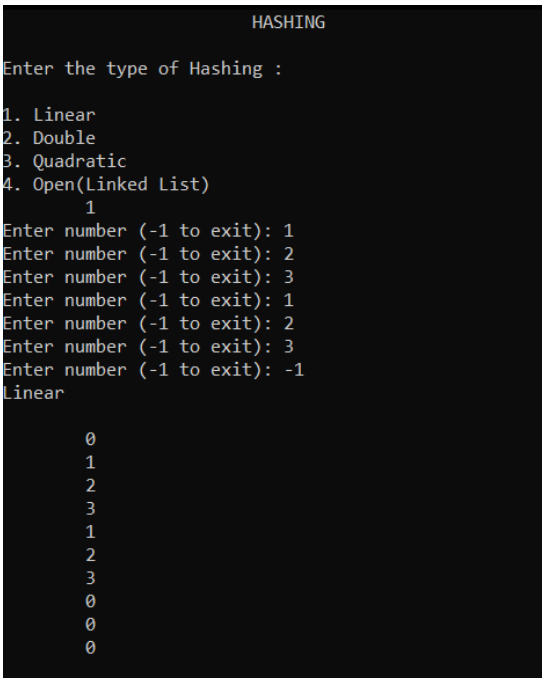
```
void Push(int value,int count)
{
	ptr=head;

	for(int i=1;i<=count;i++)
	{
		ptr=ptr->next;
	}

	if(ptr->data==0)
	ptr->data=value;

	else
	{
		node* temp=new node;
		temp->data=value;
		temp->next=NULL;
	   temp->next1=NULL;

	   ptr->next1=temp;
	}
}

void showAll()
{
	cout<<endl<<endl;
	ptr=head;
	int i=0;
	while(ptr->next!=NULL)
	{
		cout<<"\t"<<i++<<"\t\t\t"<<ptr->data;

		ptr1=ptr;

		if(ptr1->next1!=NULL)
		{
			ptr1=ptr1->next1;
			while(ptr1->next1!=NULL)
	    {
			cout<<"\t"<<ptr1->data;
			ptr1=ptr1->next1;
	    }
```

```
                    cout<<"\t"<<ptr1->data;
                    }
                    cout<<endl;
                    ptr=ptr->next;
                }
        }
};

int main()
{
        node h;
        h.start();
        int a,b,opt;
        cout<<"\t\t\tHASHING\n\nEnter the type of Hashing :\n\n1. Linear\n2. Double\n3.
Quadratic\n4. Open(Linked List)\n\t";
        cin>>opt;

        if(opt==4)
        {
                cout<<"Open\n\n";
                do
                {
                    cout<<"Enter number (-1 to exit): ";
                    cin>>a;


                    b=a%10;

                    if(a!=-1)
                    h.Push(a,b);
                }
        while(a!=-1);
        h.showAll();
        }

else
        {

                int arr[10];

        for(int i=0;i<10;i++)
        {
```

```
        arr[i]=0;
}
int i=0;
do
{
        cout<<"Enter number (-1 to exit): ";
        cin>>a;


        b=a%10;

        if(opt==2)
        {
                b=(b+i*a)%10;
        }


        else if(opt==3)
        {
                b=(b+i*i)%10;
        }



        if(a!=-1)
        {
                while(arr[b]!=0)
                {
                        arr[++b];

                        if(b==10)
                        b=0;
                }

                arr[b]=a;
        }
        i++;
}
while(a!=-1);

if(opt==1)
        cout<<"Linear\n\n";
```

```
        else if(opt==2)
                cout<<"Double\n\n";

        else if(opt==3)
                cout<<"Quadratic\n\n";


        for(int i=0;i<10;i++)
        {
                cout<<"\t"<<arr[i]<<endl;
        }
        }

}
```

Output:



Results & Observations:

In this lab we have learnt about the basics and implementations of Hashing. Hashing is used to store data in a memory location and to retrieve in a constant time making time complexity less. This function helps us for faster entry and retrieval of data. We can use different formulas for different hashing tables and upon our need.