Name: Aditi Kohale

Course: C, DSA and C++

Topic: Linked List – Assignment 1

Q. Linked List of Malls:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct mall
{
        char mName[20];
        int shops;
    float revenue;
        struct mall *next;
};

struct mall *head=NULL;

void printLL()
{
        struct mall *temp=head;
        while(temp!=NULL)
        {
                printf("| %s",temp->mName);
                printf("|| %d",temp->shops);
                printf("|| %f |->",temp->revenue);
                temp=temp->next;
        }
```

```
}

struct mall *createNode()
{
	struct mall *newNode=(struct mall*) malloc(sizeof(struct mall));
	getchar();
	printf("Enter mall Name:\n");
	char ch;
	int i=0;
	while((ch=getchar())!='\n')
	{
		(*newNode).mName[i]=ch;
		i++;
	}
	printf("Enter No. of Shops:\n");
	scanf("%d",&newNode->shops);
	printf("Enter the Revenue:\n");
	scanf("%f",&newNode->revenue);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct mall *newNode=createNode();
	struct mall *temp=head;
	if(head==NULL)
	{
		head=newNode;
```

```
        }
        else
        {
                while(temp->next!=NULL)
                {
                        temp=temp->next;
                }
                temp->next=newNode;
        }
}

void main()
{
        int count;
        printf("Enter the no. of malls you want:\n");
        scanf("%d",&count);
        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printf("The malls are:\n");
        printLL();
}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ vim ques1.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ cc ques1.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ ./a.out
Enter the no. of malls you want:
3
Enter mall Name:
Empress Mall
Enter No. of Shops:
150
Enter the Revenue:
500000
Enter mall Name:
Central Mall
Enter No. of Shops:
50
Enter the Revenue:
50000
Enter mall Name:
VR Mall
Enter No. of Shops:
100
Enter the Revenue:
1000000
The malls are:
| Empress Mall|| 150|| 500000.000000 |->| Central Mall|| 50|| 50000.000000 |->| VR Mall|| 100|| 1000000.000000 |->aditi@
DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$
```

Q.2. Linked List of States in India:

Code:

```
#include<stdio.h>

#include<stdlib.h>


struct state

{

        char sName[20];

        float popul;

    float budget;

        float literacy;

        struct state *next;

};


struct state *head=NULL;


void printLL()

{
```

```
	struct state *temp=head;
	int i=1;
	while(temp!=NULL)
	{
		printf("State%d:\n",i++);
		printf("|State Name: %s|\n",temp->sName);
		printf("|Population: %f|\n",temp->popul);
		printf("|Budget: %f|\n",temp->budget);
		printf("|Literacy: %f| -->\n",temp->literacy);
		temp=temp->next;
	}
}

struct state *createNode()
{
	struct state *newNode=(struct state*) malloc(sizeof(struct state));
	getchar();
	printf("Enter State Name:\n");
	char ch;
	int i=0;
	while((ch=getchar())!='\n')
	{
		(*newNode).sName[i]=ch;
		i++;
	}
	printf("Enter Population:\n");
	scanf("%f",&newNode->popul);
	printf("Enter Budget:\n");
	scanf("%f",&newNode->budget);
```

```
	printf("Enter Literacy:\n");
	scanf("%f",&newNode->literacy);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct state *newNode=createNode();
	struct state *temp=head;
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
{
	int count;
	printf("Enter the no. of states you want:\n");
	scanf("%d",&count);
```

```
        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printf("The states are:\n");
        printLL();
}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ vim ques2.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ ./a.out
Enter the no. of states you want:
4
Enter State Name:
Maharashtra
Enter Population:
13.39
Enter Budget:
6.12
Enter Literacy:
82.33
Enter State Name:
Goa
Enter Population:
15.96
Enter Budget:
2.68
Enter Literacy:
100
Enter State Name:
Haryana
Enter Population:
30.20
Enter Budget:
20.04
Enter Literacy:
75.55
```

```
Enter State Name:
Uttar Pradesh
Enter Population:
246.2
Enter Budget:
61.09
Enter Literacy:
73
The states are:
Mall1:
|State Name: Maharashtra|
|Population: 13.390000|
|Budget: 6.120000|
|Literacy: 82.330002| -->
Mall2:
|State Name: Goa|
|Population: 15.960000|
|Budget: 2.680000|
|Literacy: 100.000000| -->
Mall3:
|State Name: Haryana|
|Population: 30.200001|
|Budget: 20.040001|
|Literacy: 75.550003| -->
Mall4:
|State Name: Uttar Pradesh|
|Population: 246.199997|
|Budget: 61.090000|
|Literacy: 73.000000| -->
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$
```

Q.3. Linked List of Festivals in India:

Code:

```
#include<stdio.h>

#include<stdlib.h>


struct festival

{

        char fName[20];

        struct festival *next;

};


struct festival *head=NULL;


void printLL()
```

```
{
	struct festival *temp=head;
	while(temp!=NULL)
	{
		printf("| %s | -->",temp->fName);
		temp=temp->next;
	}
}

struct festival *createNode()
{
	struct festival *newNode=(struct festival*) malloc(sizeof(struct festival));
	printf("Enter Festival Name:\n");
	char ch;
	int i=0;
	while((ch=getchar())!='\n')
	{
		(*newNode).fName[i]=ch;
		i++;
	}
	(*newNode).fName[i]='\0';
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct festival *newNode=createNode();
	struct festival *temp=head;
```

```
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
{
	int count;
	printf("No. of Festivals you want to enter:\n");
	scanf("%d",&count);
	getchar();
	for(int i=1;i<=count;i++)
	{
		addNode();
	}
	printf("The festivals are:\n");
	printLL();
}
```

Output:

Q. Count No. of Nodes in above example:

Code:

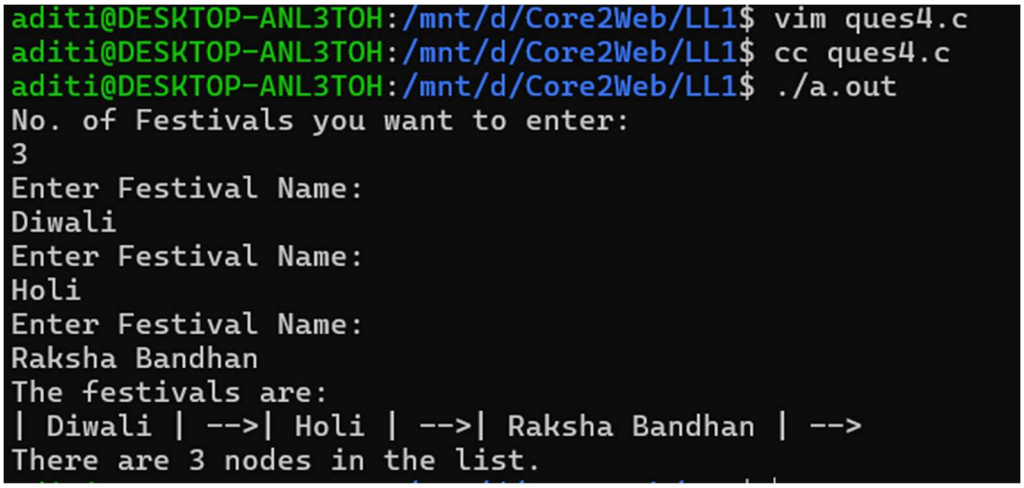
```
#include<stdio.h>

#include<stdlib.h>


struct festival

{

        char fName[20];

        struct festival *next;

};


struct festival *head=NULL;


void printLL()

{

        struct festival *temp=head;

        while(temp!=NULL)

        {
```

```
                printf("| %s | -->",temp->fName);
                temp=temp->next;
        }
        printf("\n");
}

struct festival *createNode()
{
        struct festival *newNode=(struct festival*) malloc(sizeof(struct festival));
        printf("Enter Festival Name:\n");
        char ch;
        int i=0;
        while((ch=getchar())!='\n')
        {
                (*newNode).fName[i]=ch;
                i++;
        }
        (*newNode).fName[i]='\0';
        newNode->next=NULL;
        return newNode;
}

void addNode()
{
        struct festival *newNode=createNode();
        struct festival *temp=head;
        if(head==NULL)
        {
                head=newNode;
```

```
        }
        else
        {
                while(temp->next!=NULL)
                {
                        temp=temp->next;
                }
                temp->next=newNode;
        }
}

int countNodes()
{
        int count=0;
        struct festival *temp=head;
        while(temp!=NULL)
        {
         count++;
                temp=temp->next;
        }
        return count;
}

void main()
{
        int num;
        printf("No. of Festivals you want to enter:\n");
        scanf("%d",&num);
        getchar();
```

```
        for(int i=1;i<=num;i++)
        {
                addNode();
        }
        printf("The festivals are:\n");
        printLL();
        int count = countNodes();
        printf("There are %d nodes in the list.\n",count);
}
```

Output:



Q. 5. Demo Structure with integer data. Take the number of nodes from the user and print the addition of the data:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct Demo
{
	int data;
	struct Demo *next;
};

struct Demo *head=NULL;

int sumNode()
{
	int sum=0;
	struct Demo *temp=head;
	if(head==NULL)
	{
		printf("Linked List is Empty.\n");
		return 0;
	}
	else
	{
		while(temp!=NULL)
		{
			sum=sum+temp->data;
			temp=temp->next;
```

```
                }
                return sum;
        }
}

void printLL()
{
        printf("The Linked list is:\n");
        struct Demo *temp=head;
        while(temp!=NULL)
        {
                printf("%d->",temp->data);
                temp=temp->next;
        }
        printf("\n");
}



struct Demo *createNode()
{
        struct Demo *newNode=(struct Demo*)malloc(sizeof(struct Demo));
        printf("Enter data:\n");
        scanf("%d",&newNode->data);
        newNode->next=NULL;
        return newNode;
}

void addNode()
{
```

```
	struct Demo *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct Demo *temp=head;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
{
	int count;
	printf("Enter the no. of nodes:\n");
	scanf("%d",&count);

	for(int i=1;i<=count;i++)
	{
		addNode();
	}
	printLL();
	int sum=sumNode();
	printf("The sum of Nodes is %d\n",sum);
```

}

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ vim ques5.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ cc ques5.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ ./a.out
Enter the no. of nodes:
5
Enter data:
10
Enter data:
20
Enter data:
30
Enter data:
40
Enter data:
50
The Linked list is:
10->20->30->40->50->
The sum of Nodes is 150
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$
```

Q.6. Print the Addition of the first and the last node from the above code:

Code:

```
#include<stdio.h>

#include<stdlib.h>


struct Demo

{

        int data;

        struct Demo *next;

};


struct Demo *head=NULL;


int countNode()

{

        int count=0;

        if(head==NULL)
```

```
	{
		return 0;
	}
	else
	{
		struct Demo *temp=head;
		while(temp!=NULL)
		{
		    count++;
			temp=temp->next;
		}
		return count;
	}
}

int sumNode()
{
	int count=countNode();
	int sum=0;
	if (head==NULL)
	{
		printf("The linked list is empty\n");
		return 0;
	}
	else
	{
		if(count==1)
		{
			return head->data;
```

```
                }
                else
                {
                        struct Demo *temp=head;
                        sum=temp->data;
                        while(temp->next!=NULL)
                        {
                                temp=temp->next;
                        }
                        sum=sum+temp->data;
                        return sum;
                }

        }
}


void printLL()
{
        printf("The Linked list is:\n");
        struct Demo *temp=head;
        while(temp!=NULL)
        {
                printf("%d->",temp->data);
                temp=temp->next;
        }
        printf("\n");
}
```

```
struct Demo *createNode()
{
	struct Demo *newNode=(struct Demo*)malloc(sizeof(struct Demo));
	printf("Enter data:\n");
	scanf("%d",&newNode->data);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct Demo *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct Demo *temp=head;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
```

```
{
        int count;
        printf("Enter the no. of nodes:\n");
        scanf("%d",&count);

        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printLL();
        int sum=sumNode();
        printf("The sum of First and Last Nodes is %d\n",sum);
}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ vim ques6.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ cc ques6.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ ./a.out
Enter the no. of nodes:
5
Enter data:
10
Enter data:
20
Enter data:
30
Enter data:
40
Enter data:
50
The Linked list is:
10->20->30->40->50->
The sum of First and Last Nodes is 60
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$
```

Q.7. Print the maximum integer from the above nodes:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct Demo
{
	int data;
	struct Demo *next;
};

struct Demo *head=NULL;

int countNode()
{
	int count=0;
	if(head==NULL)
	{
		return 0;
	}
	else
	{
		struct Demo *temp=head;
		while(temp!=NULL)
		{
		    count++;
			temp=temp->next;
		}
```

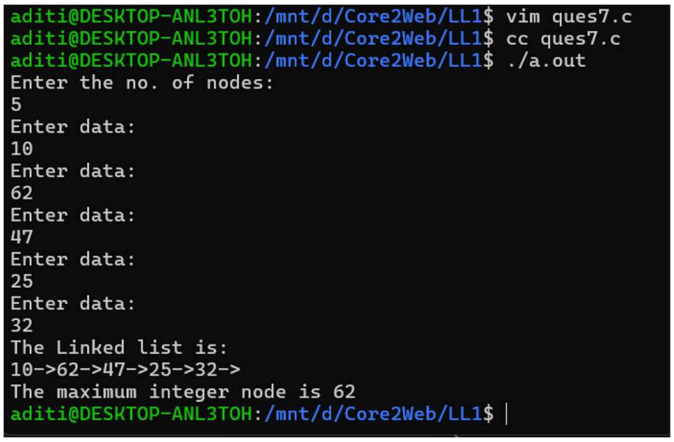
```
                return count;
        }
}

int maxNode()
{
        int count=countNode();
        if(head==NULL)
        {
                printf("Linked list is empty.\n");
                return -1;
        }
        else
        {
                if(count==1)
                {
                        return head->data;
                }
                else
                {
                        struct Demo *temp=head;
                        int max=temp->data;
                        while(temp!=NULL)
                        {
                                if(temp->data>max)
                                {
                                        max=temp->data;
                                }
                                temp=temp->next;
```

```
                        }
                        return max;
                }
        }
}



void printLL()
{
        printf("The Linked list is:\n");
        struct Demo *temp=head;
        while(temp!=NULL)
        {
                printf("%d->",temp->data);
                temp=temp->next;
        }
        printf("\n");
}



struct Demo *createNode()
{
        struct Demo *newNode=(struct Demo*)malloc(sizeof(struct Demo));
        printf("Enter data:\n");
        scanf("%d",&newNode->data);
        newNode->next=NULL;
        return newNode;
}
```

```
void addNode()
{
	struct Demo *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct Demo *temp=head;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
{
	int count;
	printf("Enter the no. of nodes:\n");
	scanf("%d",&count);

	for(int i=1;i<=count;i++)
	{
		addNode();
	}
```

```
	printLL();
	int max=maxNode();
	printf("The maximum integer node is %d\n",max);
}
```

Output:



Q.8. Print the minimum node from the above example:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct Demo
{
	int data;
	struct Demo *next;
};

struct Demo *head=NULL;
```

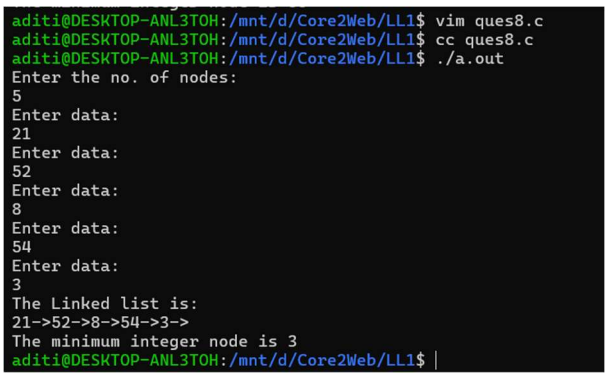
```
int countNode()
{
	int count=0;
	if(head==NULL)
	{
		return 0;
	}
	else
	{
		struct Demo *temp=head;
		while(temp!=NULL)
		{
		    count++;
			temp=temp->next;
		}
		return count;
	}
}

int minNode()
{
	int count=countNode();
	if(head==NULL)
	{
		printf("Linked list is empty.\n");
		return -1;
	}
	else
```

```
	{
		if(count==1)
		{
			return head->data;
		}
		else
		{
			struct Demo *temp=head;
			int min=temp->data;
			while(temp!=NULL)
			{
				if(temp->data<min)
				{
					min=temp->data;
				}
				temp=temp->next;
			}
			return min;
		}
	}
}



void printLL()
{
	printf("The Linked list is:\n");
	struct Demo *temp=head;
	while(temp!=NULL)
```

```
	{
		printf("%d->",temp->data);
		temp=temp->next;
	}
	printf("\n");
}


struct Demo *createNode()
{
	struct Demo *newNode=(struct Demo*)malloc(sizeof(struct Demo));
	printf("Enter data:\n");
	scanf("%d",&newNode->data);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct Demo *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct Demo *temp=head;
		while(temp->next!=NULL)
		{
```

```
                temp=temp->next;
            }
            temp->next=newNode;
        }
}

void main()
{
        int count;
        printf("Enter the no. of nodes:\n");
        scanf("%d",&count);

        for(int i=1;i<=count;i++)
        {
            addNode();
        }
        printLL();
        int min=minNode();
        printf("The minimum integer node is %d\n",min);
}
```

Output:



Q.9. Check the prime number present in the above nodes:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct Demo
{
	int data;
	struct Demo *next;
};

struct Demo *head=NULL;

int countNode()
{
	int count=0;
	if(head==NULL)
	{
		return 0;
	}
	else
	{
		struct Demo *temp=head;
		while(temp!=NULL)
		{
		    count++;
			temp=temp->next;
		}
```

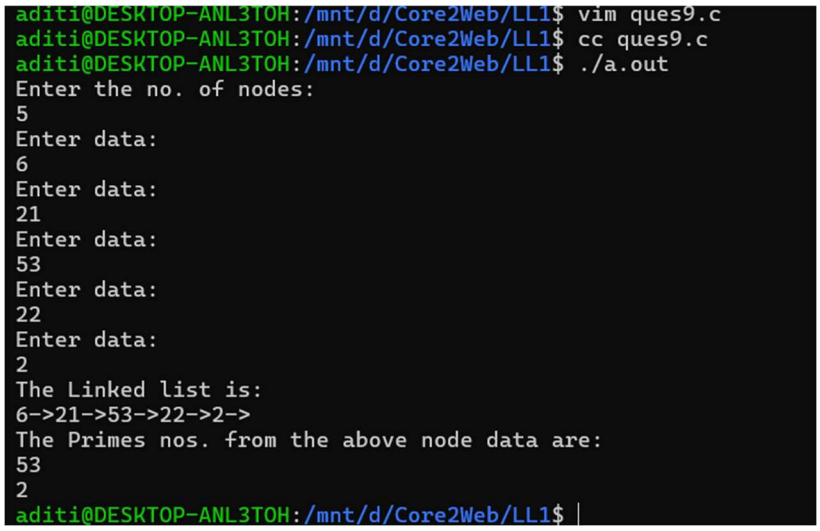
```
                return count;
        }
}

void primeNode()
{
        int count=countNode();
        if(head==NULL)
        {
                printf("The linked list is empty, hence no Prime Numbers\n");
        }
        else
        {
                struct Demo *temp=head;
                while(temp!=NULL)
                {
                        if(temp->data==0 || temp->data==1)
                        {
                                printf("%d is neither prime nor composite\n",temp->data);
                        }
                        else
                        {
                                int flag=0;
                                for(int i=2;i*i<=temp->data;i++)
                                {
                                        if(temp->data%i==0)
                                        {
                                                flag=1;
                                                break;
```

```
                                        }
                                }
                                if(flag==0)
                                {
                                        printf("%d\n",temp->data);
                                }
                        }
                        temp=temp->next;
                }
        }
}




void printLL()
{
        printf("The Linked list is:\n");
        struct Demo *temp=head;
        while(temp!=NULL)
        {
                printf("%d->",temp->data);
                temp=temp->next;
        }
        printf("\n");
}



struct Demo *createNode()
```

```
{
	struct Demo *newNode=(struct Demo*)malloc(sizeof(struct Demo));
	printf("Enter data:\n");
	scanf("%d",&newNode->data);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct Demo *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct Demo *temp=head;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void main()
{
	int count;
```

```
        printf("Enter the no. of nodes:\n");
        scanf("%d",&count);

        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printLL();
        printf("The Primes nos. from the above node data are:\n");
        primeNode();
}
```

Output:



Q.10. Write a real-time example of linked list and print its data. Take 5 nodes from the user:

Code:

```
#include<stdio.h>
#include<stdlib.h>

struct menu
{
	char name[20];
	float price;
	struct menu *next;
};

struct menu *head=NULL;

struct menu *createNode()
{
	struct menu *newNode=(struct menu*)malloc(sizeof(struct menu));
	printf("Enter the dish name:\n");
	char ch;
	int i=0;
	getchar();
	while((ch=getchar())!='\n')
	{
		(*newNode).name[i]=ch;
		i++;
	}
	(*newNode).name[i]='\0';
	printf("Enter the price of the dish\n");
```

```
	scanf("%f",&newNode->price);
	newNode->next=NULL;
	return newNode;
}

void addNode()
{
	struct menu *newNode=createNode();
	if(head==NULL)
	{
		head=newNode;
	}
	else
	{
		struct menu *temp=head;
		while(temp->next!=NULL)
		{
			temp=temp->next;
		}
		temp->next=newNode;
	}
}

void printLL()
{
	struct menu *temp=head;
	while(temp!=NULL)
	{
		printf("|%s||%f|->",temp->name,temp->price);
```

```
            temp=temp->next;
        }
}

void main()
{
        int count;
        printf("Enter the no. of nodes:\n");
        scanf("%d",&count);
        for(int i=1;i<=count;i++)
        {
            addNode();
        }
        printf("The Menu is:\n");
        printLL();
}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ vim ques10.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ cc ques10.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL1$ ./a.out
Enter the no. of nodes:
5
Enter the dish name:
Dosa
Enter the price of the dish
70
Enter the dish name:
Idli
Enter the price of the dish
60
Enter the dish name:
Paratha
Enter the price of the dish
50
Enter the dish name:
Upma
Enter the price of the dish
40
Enter the dish name:
Pohe
Enter the price of the dish
40
The Menu is:
|Dosa||70.000000|->|Idli||60.000000|->|Paratha||50.000000|->|Upma||40.000000|->|Pohe||40.000000|-
/mnt/d/Core2Web/LL1$
```