Name: Aditi Kohale

Course: C, DSA and C++

Topic: Linked List – Assignment 2

Q. Find the first occurrence of a number:

Code:

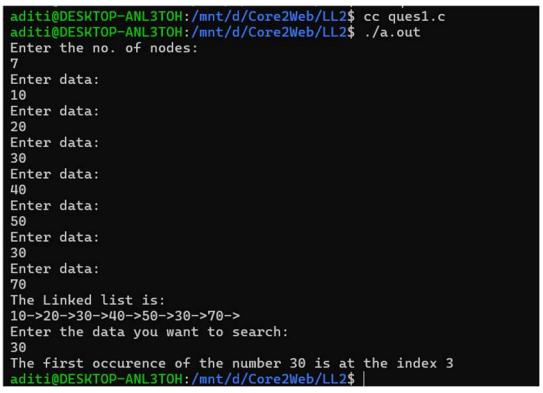
```
#include<stdio.h>
#include<stdlib.h>

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

struct Demo *head=NULL;

int first_occurence(int num)
{
	if(head==NULL)
	{
		return -1;
	}
	else
	{
		int count=0;
		int flag=0;
		struct Demo *temp=head;
```

```
		while(temp!=NULL)
		{
			count++;
			if(temp->data==num)
			{
				flag=1;
				break;
			}
			temp=temp->next;
		}
		if(flag==0)
		{
			return -1;
		}
		else
		{
			return count;
		}
	}
}

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 num;
        printf("Enter the data you want to search:\n");
        scanf("%d",&num);
        int occ=first_occurence(num);
        printf("The first occurence of the number %d is at the index %d\n",num,occ);

}
```

Output:



Q.2. To find the second last occurrence of the number:

Code:

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

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

struct Demo *head=NULL;

void secondLast_occ(int num)
{
	if(head==NULL)
	{
		printf("Linked list empty\n");
	}
	else
	{
		struct Demo *temp=head;
		int count=1;
		int last=-1;
		int secondLast=-1;
		while(temp!=NULL)
		{
			if(temp->data==num)
```

```
                {
                        secondLast=last;
                        last=count;
                }
                temp=temp->next;
                count++;
        }
        if(secondLast!=-1)
        {
                printf("The second last occurence of %d is found at %d\n",num,secondLast);
        }
        else if(last!=-1)
        {
                printf("Not second last but last occurence of %d was found at %d\n",num,last);
        }
        else
        {
                printf("No occurence found\n");
        }
    }
}

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 num;
        printf("Enter the data you want to search:\n");
        scanf("%d",&num);
        secondLast_occ(num);

}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ vim ques2.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ cc ques2.c
^[[Aaditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ ./a.out
Enter the no. of nodes:
7
Enter data:
10
Enter data:
20
Enter data:
30
Enter data:
40
Enter data:
30
Enter data:
30
Enter data:
70
The Linked list is:
10->20->30->40->30->30->70->
Enter the data you want to search:
30
The second last occurence of 30 is found at 5
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$
```

Q.3. WAP that searches the occurrences of the data in the linked list:

Code:

```
#include<stdio.h>

#include<stdlib.h>


struct Demo

{

        int data;

        struct Demo *next;

};


struct Demo *head=NULL;


int occurence(int num)

{

        if(head==NULL)

        {
```

```
		return -1;
	}
	else
	{
		int count=0;
		int flag=0;
		struct Demo *temp=head;
		while(temp!=NULL)
		{
			if(temp->data==num)
			{
				count++;
				flag=1;
			}
			temp=temp->next;
		}
		if(flag==0)
		{
			return -1;
		}
		else
		{
			return count;
		}
	}
}

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 num;
        printf("Enter the data you want to search:\n");
        scanf("%d",&num);
        int occ=occurence(num);
        printf("There are %d occurences of the number %d\n",occ,num);

}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ vim ques3.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ cc ques3.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ ./a.out
Enter the no. of nodes:
7
Enter data:
10
Enter data:
20
Enter data:
30
Enter data:
40
Enter data:
50
Enter data:
30
Enter data:
70
The Linked list is:
10->20->30->40->50->30->70->
Enter the data you want to search:
30
There are 2 occurences of the number 30
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$
```

Q.4. WAP that adds the digits of the data at each node:

Code:

```
#include<stdio.h>

#include<stdlib.h>


struct Demo

{

        int data;

        struct Demo *next;

};


struct Demo *head=NULL;


void addDigits()

{

        if(head==NULL)

        {

                printf("Linked list is empty\n");
```

```
	}
	else
	{
		struct Demo *temp=head;
		while(temp!=NULL)
		{
			int org=temp->data;
			int last=0;
			int sum=0;
			while(org!=0)
			{
				last=org%10;
				sum=sum+last;
				org=org/10;
			}
			temp->data=sum;
			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();
        addDigits();
        printf("After addition of the digits the new linked list is:\n");
        printLL();

}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ ./a.out
Enter the no. of nodes:
6
Enter data:
11
Enter data:
12
Enter data:
13
Enter data:
141
Enter data:
2
Enter data:
158
The Linked list is:
11->12->13->141->2->158->
After addition of the digits the new linked list is:
The Linked list is:
2->3->4->6->2->14->
```

Q.5. WAP that seaches all the palindrome from the linked list and prints the position of the palindrome:

Code:

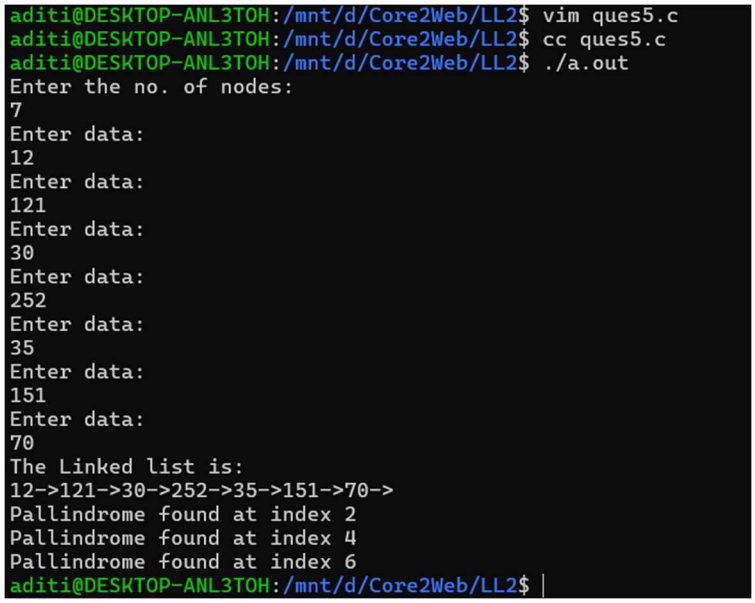
```
#include<stdio.h>
#include<stdlib.h>

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

struct Demo *head=NULL;

void checkPallindrome()
{
	if(head==NULL)
	{
		printf("Linked list is empty\n");
	}
	else
	{
		int count=1;
		struct Demo *temp=head;
		while(temp!=NULL)
		{
			int org=temp->data;
			int last=0;
			int rev=0;
```

```
                        while(org!=0)
                        {
                                last=org%10;
                                rev=(rev*10)+last;
                                org=org/10;
                        }
                        if(rev==temp->data)
                        {
                                printf("Pallindrome found at index %d\n",count);
                        }
                        temp=temp->next;
                        count++;
                }
        }
}

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();
    checkPallindrome();

}
```

Output:



Q.6. WAP that accepts a singly linear linked list from the user. Take a number from the user and print the data of that length.

Code:

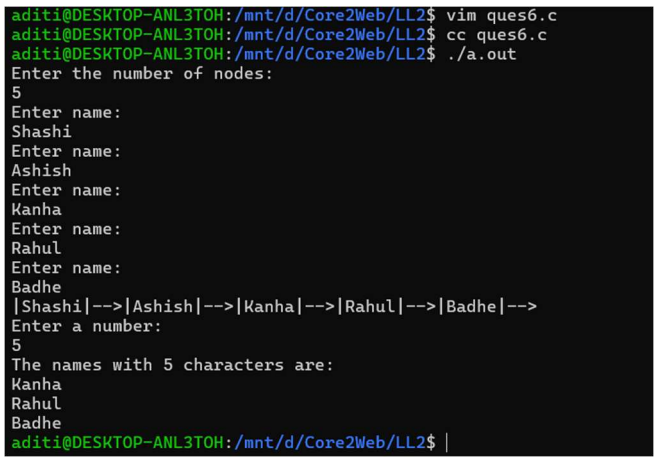
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct node
{
	char str[20];
	struct node *next;
};

struct node *head=NULL;

int mystrlen(char *str)
{
	int len=0;
	while(*str!='\0')
	{
		len++;
		str++;
	}
	return len;
}

void count_char(int num)
{
	if(head==NULL)
```

```
	{
		printf("Linked list is empty\n");
	}
	else
	{
		struct node *temp=head;
		while(temp!=NULL)
		{
			int len=mystrlen(temp->str);
			if(len==num)
			{
				printf("%s\n",temp->str);
			}
			temp=temp->next;
		}
	}
}

struct node *createNode()
{
	struct node *newNode=(struct node*)malloc(sizeof(struct node));
	printf("Enter name:\n");
	fgets(newNode->str,15,stdin);
	int len=mystrlen(newNode->str);
	if(newNode->str[len-1]=='\n')
	{
		newNode->str[len-1]='\0';
	}
	newNode->next=NULL;
```

```
	return newNode;
}

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

void printLL()
{
	struct node *temp=head;
	while(temp!=NULL)
	{
		printf("|%s|-->",temp->str);
		temp=temp->next;
	}
```

```
        printf("\n");
}

void main()
{
        int count;
        printf("Enter the number of nodes:\n");
        scanf("%d",&count);
        getchar();
        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printLL();
        printf("Enter a number:\n");
        int num;
        scanf("%d",&num);
        printf("The names with %d characters are:\n",num);
        count_char(num);
}
```

Output:



Q.7. Reverse the data elements in the above example:

Code:

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

struct node
{
	char str[20];
	struct node *next;
};

struct node *head=NULL;

int mystrlen(char *str)
{
	int len=0;
	while(*str!='\0')
	{
		len++;
		str++;
	}
	return len;
}

char *mystrrev(char *str)
{
```

```
        char *start=str;
            char *temp=str;
            while(*temp!='\0')
            {
                    temp++;
            }
            temp--;
            while(start<temp)
            {
            char tempvar=*start;
            *start=*temp;
            *temp=tempvar;
            start++;
            temp--;
            }
            return str;
}



void reverse()
{
            if(head==NULL)
            {
                    printf("Linked list is empty\n");
            }
            else
            {
```

```
                struct node *temp=head;
                while(temp!=NULL)
                {
                        strcpy(temp->str,mystrrev(temp->str));
                        temp=temp->next;
                }
        }
}


struct node *createNode()
{
        struct node *newNode=(struct node*)malloc(sizeof(struct node));
        printf("Enter name:\n");
        fgets(newNode->str,15,stdin);
        int len=mystrlen(newNode->str);
        if(newNode->str[len-1]=='\n')
        {
                newNode->str[len-1]='\0';
        }
        newNode->next=NULL;
        return newNode;
}

void addNode()
{
        struct node *newNode=createNode();
```

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

void printLL()
{
	struct node *temp=head;
	while(temp!=NULL)
	{
		printf("|%s|-->",temp->str);
		temp=temp->next;
	}
	printf("\n");
}

void main()
```

```
{
        int count;
        printf("Enter the number of nodes:\n");
        scanf("%d",&count);
        getchar();
        for(int i=1;i<=count;i++)
        {
                addNode();
        }
        printLL();
        reverse();
        printLL();
}
```

Output:

```
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ vim ques7.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ cc ques7.c
aditi@DESKTOP-ANL3TOH:/mnt/d/Core2Web/LL2$ ./a.out
Enter the number of nodes:
5
Enter name:
Shashi
Enter name:
Ashsih
Enter name:
Kanha
Enter name:
Rahul
Enter name:
Badhe
|Shashi|-->|Ashsih|-->|Kanha|-->|Rahul|-->|Badhe|-->
|ihsahS|-->|hishsA|-->|ahnaK|-->|luhaR|-->|ehdaB|-->
```

Q.8. Keep only the elements in the linked list whose length is equal to the number taken from the user:

Code:

```
#include<stdio.h>

#include<stdlib.h>
```

```
#include<string.h>

struct node
{
	char str[20];
	struct node *next;
};

struct node *head=NULL;

int mystrlen(char *str)
{
	int len=0;
	while(*str!='\0')
	{
		len++;
		str++;
	}
	return len;
}

void limit_count(int limit)
{
	if(head==NULL)
	{
		printf("Linked list is empty\n");
	}
	else
	{
```

```
        struct node *temp=head;
        struct node *prev=NULL;
        while(temp!=NULL)
        {
            int len=mystrlen(temp->str);
            if(len!=limit)
            {
                struct node *node_to_delete=temp;
                if(prev==NULL)
                {
                    head=temp->next;
                    temp=head;
                }
                else
                {
                    prev->next=temp->next;
                    temp=temp->next;
                }
                free(node_to_delete);
            }
            else
            {
                prev=temp;
                temp=temp->next;
            }
        }
    }
}
```

```
struct node *createNode()
{
	struct node *newNode=(struct node*)malloc(sizeof(struct node));
	printf("Enter name:\n");
	fgets(newNode->str,15,stdin);
	int len=mystrlen(newNode->str);
	if(newNode->str[len-1]=='\n')
	{
		newNode->str[len-1]='\0';
	}
	newNode->next=NULL;
	return newNode;
}

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

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

void printLL()
{
        struct node *temp=head;
        while(temp!=NULL)
        {
                printf("|%s|-->",temp->str);
                temp=temp->next;
        }
        printf("\n");
}

void main()
{
        int count;
        printf("Enter the number of nodes:\n");
        scanf("%d",&count);
        getchar();
        for(int i=1;i<=count;i++)
        {
                addNode();
        }
```

```
        printLL();
        int limit;
        printf("Enter the limit:\n");
        scanf("%d",&limit);
        limit_count(limit);
        printLL();
}
```

Output: