Assignment on Linked List

Name: Bharat Upadhyay
2017641
PCS 302(Data Structures with C)

Q1 – Assuming we already have a linked list with pointer start. Write a function to count nodes whose info field is having data more than 10000. Value of count has to be printed outside of the function. You may either :

- **Use a return statement**
- **Without using return statement**

ALGORITHM:

Assumptions : Using struct define a datatype nodetype having first field info of type int and 2nd field next which is an address of type nodetype. Take a nodetype pointer and initialize it to NULL as initially Linked list is empty.

1) Start
2) Create a menu for the user to choose whether to insert, display, count without using return statement or with using return statement
3) Insert the values into the linked list. Using malloc allocate memory for a nodetype and assign its address to a pointer p
4) If p==NULL ,memory not allocated. Go to 8.
5) Input num and assign its value to info field of p. if start==NULL then (*start)=p and (*start)->next=NULL. Else p->next=(*start) and (*start)=p. Hence linked list is created with head pointer start.
6) For counting number of nodes whose info is greater than 10000 using return statement, use while(l!=NULL) and if(l->info>10000) then increment a counter c with each iteration. Return c
7) For counting number of nodes whose info is greater than 10000 without using return statement, use call by reference and pass the address of c as an argument. If(l->info>10000) then increment c. Print c in main().
8) Stop

PROGRAM:

```
//Bharat Upadhyay
//2017641
#include <stdio.h>
#include <stdlib.h>

typedef struct node
{
   int info;
   struct node *next;
}nodetype;

void insert(nodetype **);
void display(nodetype *);
int count(nodetype *);
void count1(nodetype *,int *);

int main()
{
   nodetype *start=NULL;
   int ch,c=0;
```

```
    do
    {
       printf("\n1 - Insert");
       printf("\n2 - Display");
       printf("\n3 - Count nodes greater than 10000 using return statement");
       printf("\n4 - Count nodes greater than 10000 without using return statement");
       printf("\nPlease enter your choice : ");
       scanf("%d",&ch);
       switch(ch)
       {
       case 1:
         insert(&start);
         break;
       case 2:
         display(start);
         break;
       case 3:
         printf("\nThe number of nodes greater than 10000 are : %d\n",count(start));
         break;
       case 4:
         count1(start,&c);
         printf("\nThe number of nodes greater than 10000 are : %d\n",c);
         break;
       default:
         printf("\nInvalid Choice\n");
         break;
       }
    }while(ch<=4);

}

void insert(nodetype **start)
{
    int num;
    printf("Enter the value to be inserted : ");
    scanf("%d",&num);
    nodetype *p=NULL;
    p=(nodetype*)malloc(sizeof(nodetype));
    if(p != NULL)
    {
       p->info=num;
       if(*start==NULL)
       {
          (*start)=p;
          (*start)->next=NULL;
       }
       else
       {
          p->next=(*start);
          (*start)=p;
```

```
        }
        printf("\nNode has been successfully added\n");
    }
    else
    {
        printf("Not Enough Memory");
    }
}

void display(nodetype *l)
{
    if(l==NULL)
      printf("\nEmpty");
    else
    {
      while(l!=NULL)
      {
         printf("%d  ",l->info);
         l=l->next;
      }
    }
}

int count(nodetype *l)
{
    int c=0;
    while(l!=NULL)
    {
        if(l->info>=10000)
        {
           c++;
        }
        l=l->next;
    }
    return c;
}

void count1(nodetype *l,int *c)
{
    *c=0;
   while(l!=NULL)
    {
        if(l->info>=10000)
        {
           *c=*c+1;
        }
        l=l->next;
    }
}
```

```
1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 1
Enter the value to be inserted : 1000000

Node has been successfully added

1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 1
Enter the value to be inserted : 20000000

Node has been successfully added

1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 1
Enter the value to be inserted : 50

Node has been successfully added

1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 3

The number of nodes greater than 10000 are : 2

1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 4

The number of nodes greater than 10000 are : 2

1 - Insert
2 - Display
3 - Count nodes greater than 10000 using return statement
4 - Count nodes greater than 10000 without using return statement
Please enter your choice : 2
50  20000000  1000000
```

Q2 – Assuming that we already have a linked list. Input a number and search for that number in the linked list. If that number is found, then update the linked list by deleting all nodes after the node where number is found.

ALGORITHM:

Assumptions : Using struct define a datatype nodetype having first field info of type int and 2nd field next which is an address of type nodetype. Take a nodetype pointer and initialize it to NULL as initially Linked list is empty.

1) Start
2) Create a menu for the user to choose whether to insert, delete or display.
3) Insert the values into the linked list. Using malloc allocate memory for a nodetype and assign its address to a pointer p
4) If p==NULL ,memory not allocated. Go to 7
5) Input num and assign its value to info field of p. if l and r both are NULL then (*r)=(*l)=p. else (*r)->next=p and (*r)=p. Hence linked list is created with head pointer l and tail as r.
6) Search for the node to be deleted. If node is found, assign todelete to the node next to the node found, and free(todelete). Assign NULL to the next field of the pointer found. Display a message if node is not found
7) Stop

PROGRAM:

```
//Bharat Upadhyay
//2017641
#include <stdio.h>
#include <stdlib.h>

typedef struct node
{
   int info;
   struct node *next;
}nodetype;

void insert(nodetype **,nodetype **);
void del(nodetype **);
void display(nodetype *);

int main()
{
   nodetype *l=NULL,*r=NULL;
   int ch;
   do
   {
      printf("\n1 - Insert");
      printf("\n2 - Input the number to be searched");
      printf("\n3 - Display");
      printf("\nPlease enter your choice : ");
      scanf("%d",&ch);
      switch(ch)
      {
      case 1:
         insert(&l,&r);
         break;
      case 2:
         del(&l);
         break;
      case 3:
         display(l);
         break;
      default:
         printf("\nInvalid Choice\n");
         break;
      }
   }while(ch<=3);
}

void insert(nodetype **l,nodetype **r)
{
   int num=0;
   nodetype *p=NULL;
```

```
    printf("Enter the number to be inserted : ");
    scanf("%d",&num);
    p=(nodetype*)malloc(sizeof(nodetype));
    if(p==NULL)
        printf("\nMemory not allocated");
    else
    {
        p->info=num;
        if(*l==NULL && *r==NULL)
            *l=*r=p;
        else
        {
            (*r)->next=p;
            *r=p;
        }
        (*r)->next=NULL;
        printf("\nNode has been successfully added\n");
    }
}

void del(nodetype **l)
{
    int num,c=0;
    nodetype *temp=NULL,*todelete=NULL;
    temp=(*l);
    printf("Enter the number to be searched : ");
    scanf("%d",&num);
    while(temp!=NULL)
    {
        if(temp->info==num)
        {
            todelete=temp->next;
            temp->next=NULL;
            free(todelete);
            printf("\nNodes after %d have been successfully deleted\n",temp->info);
            c=1;
        }
        temp=temp->next;
    }
    if(c==0)
        printf("Node not found\n");
}

void display(nodetype *l)
{
    if(l==NULL)
        printf("\nEmpty");
    else
    {
        while(l!=NULL)
```

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

```
1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 1
Enter the number to be inserted : 1

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 1
Enter the number to be inserted : 2

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 1
Enter the number to be inserted : 3

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 1
Enter the number to be inserted : 4

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 3
1  2  3  4
```

```
1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 2
Enter the number to be searched : 2

Nodes after 2 have been successfully deleted

1 - Insert
2 - Input the number to be searched
3 - Display
Please enter your choice : 3
1  2
```

Q3 – Repeat Q2 for doubly linked list.

ALGORITHM:

Assumptions : Using struct define a datatype nodetype having first field as an address of prev node and 2nd field info of type int and 3rd field as an address of type nodetype. Take a nodetype pointer and initialize it to NULL as initially Linked list is empty.

1) Start
2) Create a menu for the user to choose whether to insert, delete or display.
3) Insert the values into the linked list. Using malloc allocate memory for a nodetype and assign its address to a pointer p
4) If p==NULL ,memory not allocated. Go to 7

5) Input num and assign its value to info field of p. if l and r both are NULL then (*r)=(*l)=p and p->prev=NULL. else (*r)->next=p and p->prev=(*r) and (*r)=p. Hence doubly linked list is created with head pointer l and tail as r.
6) Search for the node to be deleted. If node is found, assign todelete to the next node and free(todelete). Assign NULL to the next field of the pointer found. Display a message if node is not found
7) Stop

PROGRAM:

```
//Bharat Upadhyay
//2017641
#include <stdio.h>
#include <stdlib.h>

typedef struct node
{
   int info;
   struct node *next,*prev;
}nodetype;

void insert(nodetype **,nodetype**);
void del(nodetype**,nodetype**);
void display(nodetype*);

int main()
{
   nodetype *l=NULL,*r=NULL;
   int ch;
   do
   {
      printf("\n1 - Insert");
      printf("\n2 - Input the number to be searched");
      printf("\n3 - Display");
      printf("\nPlease nter your choice : " );
      scanf("%d",&ch);
      switch(ch)
      {
      case 1:
         insert(&l,&r);
         break;
      case 2:
         del(&l,&r);
         break;
      case 3:
         display(l);
         break;
      default:
         printf("Invalid Choice");
         break;
      }
   }while(ch<=3);
```

```
}

void insert(nodetype **l,nodetype **r)
{
   int num;
   printf("Enter the value to be inserted : ");
   scanf("%d",&num);
   nodetype *p=NULL;
   p=(nodetype*)malloc(sizeof(nodetype));
   if(p==NULL)
      printf("\nNot Enough Memory");
   else
   {
      p->info=num;
      if(*l==NULL && *r==NULL)
      {
         (*l)=(*r)=p;
         p->prev=NULL;
      }
      else
      {
         (*r)->next=p;
         p->prev=(*r);
         (*r)=p;
      }
      (*r)->next=NULL;
      printf("\nNode has been successfully added\n");
   }
}

void del(nodetype** l,nodetype **r)
{
   nodetype *temp=NULL,*todelete=NULL;
   int num,c=0;
   printf("\nEnter the number to be searched : ");
   scanf("%d",&num);
   temp=(*l);
   while(temp!=NULL)
   {
      if(temp->info==num)
      {
         temp->next=NULL;
         todelete=temp->next;
         free(todelete);
         printf("\nNodes after %d has been successfully deleted\n",temp->info);
         c=1;
      }
      temp=temp->next;
   }
   if(c==0)
```

```
    printf("\nNode not found \n");
}

void display(nodetype *l)
{
  if(l==NULL)
  {
    printf("\nEmpty\n");
  }
  else
  {
    while(l!=NULL)
    {
      printf("%d  ",l->info);
      l=l->next;
    }
    printf("\n");
  }
}
```

```
1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 1
Enter the value to be inserted : 10

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 1
Enter the value to be inserted : 20

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 1
Enter the value to be inserted : 30

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 1
Enter the value to be inserted : 40

Node has been successfully added

1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 3
10   20   30   40
```

```
1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 2

Enter the number to be searched : 20

Nodes after 20 has been successfully deleted

1 - Insert
2 - Input the number to be searched
3 - Display
Please nter your choice : 3
10   20
```

Q4 – Assuming that we have a circular linked list with pointer p (Pointer p can be anywhere). Write a function to count the number of nodes in the linked list.

ALGORITHM:

Assumptions : Using struct define a datatype nodetype having first field info of type int and 2nd field next which is an address of type nodetype. Take a nodetype pointer and initialize it to NULL as initially Linked list is empty.

1) Start
2) Create a menu for the user to choose whether to insert, count the number of nodes or display.
3) Using malloc allocate memory for a nodetype and assign its address to a pointer p
4) If p==NULL ,memory not allocated. Go to 7
5) Int the insert function input num and assign its value to info field of p. if l is NULL then (*l)=p and l->next=p. else p->next=r->next and l->prev=p and (*l)=p. Hence doubly linked list is created with head pointer p(in the main() function).
6) To count number of nodes, in a do while loop, use nodetype pointer temp=l->next and increment c till temp!=l->next. Return c to main() and print the number of nodes.
7) Stop

PROGRAM:

```
//Bharat Upadhyay
//2017641
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
   int info;
   struct node *next;
}nodetype;

void insert(nodetype**);
int count(nodetype*);
void display(nodetype*);

int main()
{
   nodetype *p=NULL;
   int ch;
   do
   {
      printf("\n1 - Insert");
      printf("\n2 - Count the number of nodes ");
      printf("\n3 - Display");
      printf("\nPlease enter your choice : ");
      scanf("%d",&ch);
      switch(ch)
      {
      case 1:
        insert(&p);
        break;
      case 2:
        printf("\nThe total number of nodes in the linked list are : %d \n",count(p));
```

```
            break;
        case 3:
            display(p);
            break;
        default:
            printf("\nInvalid Choice");
            break;
        }
    }while(ch<=3);
}

void insert(nodetype** l)
{
    int num;
    nodetype *p=NULL;
    printf("\nEnter the value to be inserted : ");
    scanf("%d",&num);
    p=(nodetype*)malloc(sizeof(nodetype));

    if(p != NULL)
    {
        p->info=num;
        if((*l)==NULL)
        {
            (*l)=p;
            (*l)->next=p;
        }
        else
        {
            p->next=(*l)->next;
            (*l)->next=p;
            (*l) = p;
        }
    }
    else
    {
        printf("\nMemory not allocated\n");
    }
}

int count(nodetype* l)
{

    int c=0;
    if(l==NULL)
        return c;

    else
    {
        nodetype *temp=l->next;
```

```
        do
        {
           c++;
           temp=temp->next;
        }while(temp!=l->next);
    }
   return c;
}

 void display(nodetype *l)
{
    if(l!=NULL)
    {
        nodetype *temp=l->next;
        do
        {
           printf("%d  ",temp->info);
           temp=temp->next;
        }while(temp!=l->next);
        printf("\n");
    }
    else
        printf("\nEmpty\n");
}
```

```
1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 1

Enter the value to be inserted : 10

1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 1

Enter the value to be inserted : 20

1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 1

Enter the value to be inserted : 30

1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 1

Enter the value to be inserted : 40

1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 2

The total number of nodes in the linked list are : 4

1 - Insert
2 - Count the number of nodes
3 - Display
Please enter your choice : 3
10  20  30  40
```