```c
/*
Title- 1.Write a C program to print all Strong number from the given linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node* next;
};
struct node* CreateNode();
void CreateLinkedList(struct node**);
void DisplayLinkedList(struct node*);
void DisplayStrong(struct node*);

void main()
{
    struct node* first = NULL;
    int choice;

    do
    {
        printf("\n----------------************************-------------------------\n");
        printf("\n1) Create Linked List\n2) Display Linked List\n3) Print Strong 
Number\n0) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: DisplayStrong(first);
                    break;
        }
    }while(choice != 0);
}

struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        printf("enter data: ");
        scanf("%d",&(newnode->data));
        newnode->next = NULL;
    }
}
void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}
void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf(" -> %d", head->data);
        head = head->next;
    }
}
void DisplayStrong(struct node* head)
{
    int element,value,sum,fact;
    if(head == NULL)
    {
        printf("Linked List not Created....\n");
    }
    else
    {
        while(head != NULL)
        {
            element = head->data;
            sum=0;
            while(element!=0)
            {
                value=element%10;
                fact=1;
                while(value>=1)
                {
                    fact*=value;
                    value--;
                }
                sum+=fact;
                element/=10;
            }
            if(sum==head->data)
            {
                printf("%d\t",head->data);
            }
            head = head -> next;
        }
        printf("\n");
    }

}


/*
Title- 2. Write a C program to sort a singly linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/
#include<stdio.h>
#include<stdlib.h>

struct node* CreateNode();
```

```
void CreateLinkedList(struct node**);
void SortLinkedListAssending(struct node**);
void SortLinkedListDesending(struct node**);
void DisplayLinkedList(struct node*);

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



void main()
{
    struct node* first = NULL;
    int choice;
    do
    {
        printf("\n1) Create Linked List\n2) Sort In Desending Order\n3) Sort In Assending
Order\n4) Display Linked List\n5) Exit\nEnter the Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: SortLinkedListDesending(&first);
                    break;
            case 3: SortLinkedListAssending(&first);
                    break;
            case 4: DisplayLinkedList(first);
                    break;
        }
    }while(choice!=5);

}


struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("Sorry,Memory not allocated... Please close the other applications or
restart your computer and try again...\n");
    }
    else
    {
        printf("Enter the data for newnode: ");
        scanf("%d",&(newnode -> data));
        newnode -> next = NULL;
    }
    return newnode;
}


void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}

void SortLinkedListDesending(struct node** head)
{
    struct node *tempnode1=NULL,*tempnode2=NULL;
    if(*head == NULL)
    {
        printf("Linked List not Created.....\n");
    }
    else
    {
        tempnode1 = *head;
        int temp;
        while(tempnode1 != NULL)
        {
            tempnode2 = tempnode1 -> next;
            while(tempnode2 != NULL)
            {
                if((tempnode1->data) < (tempnode2->data))
                {
                    temp = tempnode1->data;
                    tempnode1->data = tempnode2->data;
                    tempnode2->data = temp;
                }
                tempnode2 = tempnode2->next;
            }
            tempnode1 = tempnode1->next;
        }
    }
}

void SortLinkedListAssending(struct node** head)
{
    struct node *tempnode1=NULL,*tempnode2=NULL;
    if(*head == NULL)
    {
        printf("Linked List not Created.....\n");
    }
    else
    {
        tempnode1 = *head;
        int temp;
        while(tempnode1 != NULL)
        {
            tempnode2 = tempnode1 -> next;
            while(tempnode2 != NULL)
            {
                if((tempnode1->data) > (tempnode2->data))
                {
                    temp = tempnode1->data;
                    tempnode1->data = tempnode2->data;
                    tempnode2->data = temp;
                }
                tempnode2 = tempnode2->next;
            }
            tempnode1 = tempnode1->next;
        }
    }
```

```c
}


void DisplayLinkedList(struct node* tempnode)
{
    printf("Your Linked List is: ");
    while(tempnode != NULL)
    {
        printf(" -> %d",tempnode->data);
        tempnode = tempnode->next;
    }
}


/*
Title- 3. Write a C program to find second highest element in singly linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node* next;
};
struct node* CreateNode();
void CreateLinkedList(struct node**);
void DisplayLinkedList(struct node*);
void SecondHeighest(struct node*);

void main()
{
    struct node* first = NULL;
    int choice;

    do
    {
        printf("\n----------------************************-------------------------\n");
        printf("\n1) Create Linked List\n2) Display Linked List\n3) Second Heighest 
Element\n0) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: SecondHeighest(first);
                    break;
        }
    }while(choice != 0);
}

struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        newnode->next = NULL;
    }
}
void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(newnode != NULL)
    {
        printf("enter data: ");
        scanf("%d",&(newnode->data));
    }
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}
void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf(" -> %d", head->data);
        head = head->next;
    }
}
void SecondHeighest(struct node* head)
{
    struct node *tempnode1=NULL, *tempnode2 = NULL, *tempnode = NULL;
    int temp;
    if(head == NULL)
    {
        printf("Linked List not Created.....\n");
    }
    else if(head->next == NULL)
    {
        printf("Only one element in linked list...\n");
    }
    else
    {
        while(head != NULL)
        {
            tempnode1 = CreateNode();
            if(tempnode1!= NULL)
            {
                tempnode1 -> data = head -> data;
            }
            if(tempnode2 == NULL)
            {
                tempnode2 = tempnode1;
            }
            else
            {
                tempnode = tempnode2;
                while(tempnode -> next != NULL)
                {
                    tempnode = tempnode->next;
                }
                tempnode->next = tempnode1;
            }
            head = head->next;
        }
        head = tempnode2;
        while(head != NULL)
        {
            tempnode = head->next;
            while(tempnode != NULL)
            {
                if(tempnode->data > head->data)
                {
                    temp =tempnode->data;
                    tempnode->data = head -> data;
                    head->data = temp;
                }
                tempnode = tempnode->next;
            }
            head = head->next;
        }
        head = tempnode2;
        for(temp = 1;temp<2;temp++)
        {
            head = head->next;
        }
        printf("Second Heighest element in Linked List is: %d\n", head->data);
    }
    while(tempnode2 != NULL)
    {
        tempnode1 = tempnode2;
        tempnode2 = tempnode2->next;
        free(tempnode1);
        tempnode1 = NULL;
    }
    tempnode2 = head = tempnode = NULL;
}


/*
Title- 4. Write a C program to print all armstrong numbers from given singly linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node* next;
};
struct node* CreateNode();
void CreateLinkedList(struct node**);
void DisplayLinkedList(struct node*);
void PrintArmstrong(struct node*);

void main()
{
    struct node* first = NULL;
    int choice;

    do
    {
        printf("\n---------------************************-------------------------\n");
        printf("\n1) Create Linked List\n2) Display Linked List\n3) Print Armstrong
```

```c
Number\n0) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: PrintArmstrong(first);
                    break;
        }
    }while(choice != 0);
}

struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        printf("enter data: ");
        scanf("%d",&(newnode->data));
        newnode->next = NULL;
    }
}
void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}
void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf(" -> %d", head->data);
        head = head->next;
    }
}
void PrintArmstrong(struct node* head)
{
    int temp1,value,count,sum,mult,i;
    if(head == NULL)
    {
        printf("Linked List Not Created.....\n");
    }
    else
    {
        while(head != NULL)
        {
            // to find the no. of digits in a number
            count = 0;
            temp1 = head->data;
            while(temp1!=0)
            {
                temp1/=10;
                count++;
            }
        //      to find the armstrong value.
            sum=0;
            temp1 = head->data;
            while(temp1!=0)
            {
                value=temp1%10;
                mult=1;
                for(i=1;i<=count;i++)
                {
                    mult*=value;
                }
                sum+=mult;
                temp1/=10;
            }
            if(head->data == sum)
            {
                printf("%d\t",sum);
            }
            head = head -> next;
            }
            printf("\n");
    }
}


/*
Title- 5. Write a C program to sort only even data using singly linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/


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

struct node* CreateNode();
void CreateLinkedList(struct node**);
void EvenSeperator(struct node*, struct node**);
void DisplayLinkedList(struct node*);

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



void main()
{
    struct node *first = NULL, *even = NULL;
    int choice;
    do
    {
        printf("\n1) Create Linked List\n2) Display Original Linked List\n3) Seperate 
Even\n4) Display Even Linked List\n5) Exit\nEnter the Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: EvenSeperator(first, &even);
                    break;
            case 4: DisplayLinkedList(even);
                    break;
        }
    }while(choice!=5);

}


struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("Sorry,Memory not allocated... Please close the other applications or 
restart your computer and try again...\n");
    }
    else
    {
        newnode -> next = NULL;
    }
    return newnode;
}


void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(newnode != NULL)
    {
        printf("Enter the value for newnode: ");
        scanf("%d",&(newnode->data));
    }
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}

void EvenSeperator(struct node* head, struct node** even)
{
    struct node* tempnode = NULL;
    while(*even != NULL)
    {
        tempnode = (*even)->next;
        free(*even);
        *even = tempnode;
    }
    if(head == NULL)
    {
        printf("List List not available....\n");
    }
    else
    {
        while(head != NULL)
        {

            if((head->data)%2 == 0)
            {
                if(*even == NULL)
                {
                    *even = CreateNode();
                    (*even)->data = head->data;
                    (*even)->next = NULL;
                }
                else
                {
                    tempnode = *even;
                    while(tempnode->next != NULL)
                    {
                        tempnode = tempnode->next;
                    }
                    tempnode->next = CreateNode();
                    tempnode->next->data = head->data;
                    tempnode->next->next = NULL;
                }
            }

            head = head->next;
        }
    }


}

void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf("%d -> ",head->data);
        head = head->next;
    }
}



/*
Title- 6. Write a C program to accept string with multiple spaces and print count of
number character from string in given Singly linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
struct node
{
    char data;
    struct node* next;
};
```

```c
struct node* CreateNode();
void AcceptString(struct node**);
void DisplayLinkedList(struct node*);
void NumberCount(struct node*);

void main()
{
    struct node* first = NULL;
    int choice;

    do
    {
        printf("\n----------------************************-------------------------\n");
        printf("\n1) Accept String\n2) Display Linked List\n3) Count NUmber 
Characters\n0) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: AcceptString(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: NumberCount(first);
                    break;
        }
    }while(choice != 0);
}


struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        newnode->next = NULL;
    }
}


void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf("%c", head->data);
        head = head->next;
    }
}


void AcceptString(struct node** head)
{
    struct node *newnode = NULL, *tempnode = NULL;
    printf("Enter the String: ");
    scanf(" ");
    do
    {
        newnode = CreateNode();
        if( newnode != NULL)
        {
            scanf("%c",&(newnode->data));
        }
        if(*head == NULL)
        {
            *head = newnode;
        }
        else
        {
            tempnode = *head;
            while(tempnode->next!= NULL)
            {
                tempnode = tempnode->next;
            }
            tempnode->next = newnode;
        }
    }while((newnode->data) != '\n');
}


void NumberCount(struct node* head)
{
    int count = 0;
    while(head != NULL)
    {
        if(head->data >= 48 && head->data <= 57 )
        {
            count++;
        }
        head = head->next;
    }
    printf(" Total Number Characters In Given String Is: %d\n",count);
}




/*
Title- 7. Write a C program to check whether strings are Anagram strings or not in given
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct node
{
    char data;
    struct node* next;
};
struct node* CreateNode();
void AcceptString(struct node**);
void DisplayLinkedList(struct node*);
void AnagramCheck(struct node*, struct node*);

void main()
{
    struct node *first = NULL, *second = NULL;
    int choice;

    do
    {
        printf("\n----------------************************------------------------\n");
        printf("\n1) Accept String\n2) Display Linked List\n3) Anagram Check\n0)
Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: printf("Enter First String: \n");
                    AcceptString(&first);
                    printf("Enter Second String: \n");
                    AcceptString(&second);
                    break;
            case 2:
                    printf("First String: ");
                    DisplayLinkedList(first);
                    printf("Second String: ");
                    DisplayLinkedList(second);
                    break;
            case 3: AnagramCheck(first, second);
                    break;
        }
    }while(choice != 0);
}


struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        newnode->next = NULL;
    }
}


void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf("%c", head->data);
        head = head->next;
    }
    printf("\n");
}


void AcceptString(struct node** head)
{
    struct node *newnode = NULL, *tempnode = NULL;
    printf("Enter the String: ");
    scanf(" ");
    do
    {
        newnode = CreateNode();
        if( newnode != NULL)
        {
            scanf("%c",&(newnode->data));
        }
        if(*head == NULL)
        {
            *head = newnode;
        }
        else
        {
            tempnode = *head;
            while(tempnode->next!= NULL)
            {
                tempnode = tempnode->next;
            }
            tempnode->next = newnode;
        }
    }while((newnode->data) != '\n');
}


void AnagramCheck(struct node* first, struct node* second)
{
    int i = 1,j;
    char temp, *str1 = NULL, *str2 = NULL;
    str1 = (char*)malloc(sizeof(char));
    *str1 = '\0';
    str2 = (char*)malloc(sizeof(char));
    *str2 = '\0';
    while(first != NULL)
    {
        i++;
        str1 = (char*)realloc(str1,sizeof(char)*i);
        *(str1+(i-2)) = first->data;
        *(str1+(i-1)) = '\0';
        first = first->next;
    }
    i=1;
    while(second != NULL)
    {
        i++;
        str2 = (char*)realloc(str2,sizeof(char)*i);
        *(str2+(i-2)) = second->data;
        *(str2+(i-1)) = '\0';
        second = second->next;
    }
    i=0;
    if(strlen(str1) == strlen(str2))
    {
        while(*(str1+i) != '\0')
        {
            j = i+1;
            while(*(str1+j) != '\0')
            {
                if(*(str1+i) > *(str1+j))
                {
                    temp =*(str1+i);
                    *(str1+i)= *(str1+j);
                    *(str1+j) = temp;
                }
                j++;
            }
            i++;
        }
        i=0;
        while(*(str2+i) != '\0')
        {
            j = i+1;
            while(*(str2+j) != '\0')
            {
                if(*(str2+i) > *(str2+j))
                {
                    temp =*(str2+i);
                    *(str2+i)= *(str2+j);
                    *(str2+j) = temp;
                }
                j++;
            }
            i++;
```

```c
        }
        i = strcmp(str1,str2);
    }
    else
    {
        i=1;
    }
    if(i==0)
    {
        printf("Strings are Anagram Strings...\n");
    }
    else
    {
        printf("Strings are not Anagram Strings...\n");
    }
}




/*
Title- 8. Write a C program to print all Magic number from the given linked list.
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node* next;
};
struct node* CreateNode();
void CreateLinkedList(struct node**);
void DisplayLinkedList(struct node*);
void PrintMagicNumber(struct node*);

void main()
{
    struct node* first = NULL;
    int choice;

    do
    {
        printf("\n----------------************************-------------------------\n");
        printf("\n1) Create Linked List\n2) Display Linked List\n3) Check MAgic 
Number\n0) Exit\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: CreateLinkedList(&first);
                    break;
            case 2: DisplayLinkedList(first);
                    break;
            case 3: PrintMagicNumber(first);
                    break;
        }
    }while(choice != 0);
}

struct node* CreateNode()
{
    struct node* newnode = NULL;
    newnode = (struct node*)malloc(sizeof(struct node));
    if(newnode == NULL)
    {
        printf("memory not allocated ..... \n");
    }
    else
    {
        printf("enter data: ");
        scanf("%d",&(newnode->data));
        newnode->next = NULL;
    }
}
void CreateLinkedList(struct node** head)
{
    struct node* newnode = NULL;
    struct node* tempnode = *head;
    newnode = CreateNode();
    if(*head == NULL)
    {
        *head = newnode;
    }
    else
    {
        while(tempnode->next != NULL)
        {
            tempnode = tempnode->next;
        }
        tempnode->next = newnode;
    }
}
void DisplayLinkedList(struct node* head)
{
    while(head != NULL)
    {
        printf(" -> %d", head->data);
        head = head->next;
    }
}

void PrintMagicNumber(struct node* head)
{
    int num, sum, rev;
    while(head != NULL)
    {
        sum=0;
        rev=0;
        num = head->data;
        while(num != 0)
        {
            sum+= (num%10);
            num/=10;
        }
        num = sum;
        while(num != 0)
        {
            rev = (rev*10)+(num%10);
            num/=10;
        }
        if((sum*rev) == head->data)
        {
            printf("%d\t",head->data);
        }
        head = head->next;
    }
    printf("\n");
}
```