**DSA ASSIGNMENT-5**

**TANISHA KARMAKAR**

**21051950**

**CSE 37**

**Q1.  Write a program to create a linked list (three nodes) and print it.**

#include<stdio.h>

struct Node {

int data;

struct Node\*next;

};

int main()

{

struct Node\*HEAD;

struct Node\*PTR;

struct Node N1, N2, N3;

HEAD=&N1;

N1.data=10;

N1.next=&N2;

N2.data=20;

N2.next=&N3;

N3.data=30;

N3.next=NULL;

PTR=HEAD;

while(PTR!=NULL)

{

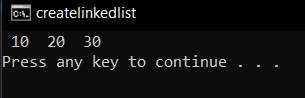
printf(" %d ", PTR->data);

PTR=PTR->next;

}

}

**Output:**



**Q2. Write a program to create a linked list and print it (dynamically).**

#include<stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\*next;

};

int main()

{

struct Node \*Head, \*New, \*ptr;

char c;

Head=NULL;

New=(struct Node\*)malloc(sizeof(struct Node));

printf("Enter the value: ");

scanf("%d", &New->data);

New->next=NULL;

Head=New;

ptr=Head;

printf("Do you want to add another node? (Y/N) ");

scanf("%c", &c);

scanf("%c", &c);

while(c=='y'||c=='Y')

{

New=(struct Node\*)malloc(sizeof(struct Node));

printf("Enter the value: ");

scanf("%d", &New->data);

New->next=NULL;

ptr->next=New;

ptr=ptr->next;

printf("Do you want to add another node? (Y/N) ");

scanf("%c", &c);

scanf("%c", &c);

}

printf("given data: \n");

ptr=Head;

while(ptr!=NULL){

printf("%d\t", ptr->data);

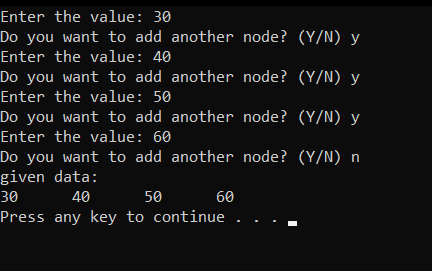
ptr=ptr->next;

}

return 0;

}

**Output:**



**Q3.  Write a program to create a LL and print its alternate nodes.**

#include<stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\*next;

};

int main()

{

struct Node \*Head, \*New, \*ptr;

char c;

Head=NULL;

New=(struct Node\*)malloc(sizeof(struct Node));

printf("Enter the value: ");

scanf("%d", &New->data);

New->next=NULL;

Head=New;

ptr=Head;

printf("Do you want to add another node? (Y/N) ");

scanf("%c", &c);

scanf("%c", &c);

while(c=='y'||c=='Y')

{

New=(struct Node\*)malloc(sizeof(struct Node));

printf("Enter the value: ");

scanf("%d", &New->data);

New->next=NULL;

ptr->next=New;

ptr=ptr->next;

printf("Do you want to add another node? (Y/N) ");

scanf("%c", &c);

scanf("%c", &c);

}

printf("given data: \n");

ptr=Head;

while(ptr!=NULL){

printf("%d\t", ptr->data);

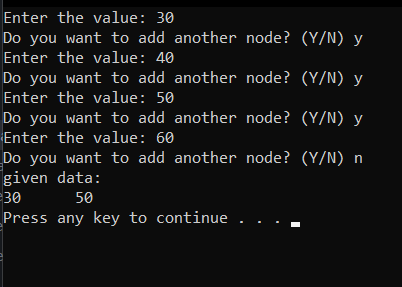
ptr=(ptr->next)->next;

}

return 0;

}

**Output:**



**Q4. WAP to Create a linked list and insert a new node as per the users choice.**

//Q4 Insert

#include <stdio.h>

#include <stdlib.h>

struct Node{

int data;

struct Node\* next;

};

void display(struct Node\* ptr){

while(ptr != NULL){

printf("%d ", ptr->data);

ptr = ptr->next;

}

}

struct Node\* insertbegin(struct Node\* HEAD){

int new;

printf ("Enter the element to insert in the begining: ");

scanf ("%d", &new);

struct Node\* ptr = (struct Node\*)malloc(sizeof(struct Node));

ptr->next = HEAD;

ptr->data = new;

return ptr;

}

struct Node\* insertAtEnd(struct Node\* HEAD){

int new;

printf("Enter the element to insert at the end: ");

scanf ("%d", &new);

struct Node\* ptr = (struct Node\*)malloc(sizeof(struct Node));

struct Node\* p = HEAD;

while (p->next!=NULL)

{

p = p->next;

}

ptr->data = new;

p->next = ptr;

ptr->next = NULL;

return HEAD;

}

struct Node\* insertAtPos(struct Node\* HEAD){

int pos, new;

printf ("Enter the element to insert: ");

scanf ("%d", &new);

printf ("Enter in which position: ");

scanf ("%d", &pos);

struct Node\* ptr = (struct Node\*)malloc(sizeof(struct Node));

struct Node\* p = HEAD;

int i = 0;

while (i != pos-2)

{

p = p->next;

i++;

}

ptr->data = new;

ptr->next = p->next;

p->next = ptr;

return HEAD;

}

int main()

{

struct Node \*HEAD;

struct Node \*NEW;

struct Node \*PTR;

char choice;

HEAD = NULL;

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

HEAD = NEW;

PTR = HEAD;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

while (choice == 'Y' || choice == 'y')

{

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

PTR->next = NEW;

PTR = PTR->next;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

}

PTR = HEAD;

printf("\nPrinting the List: \n");

while (PTR != NULL)

{

printf("%d ", PTR->data);

PTR = PTR->next;

}

printf("\n\n");

printf ("1. Insert in the beginning\n");

printf ("2. Insert at the end\n");

printf ("3. Insert at a specific position\n");

printf ("\nEnter your choice: ");

scanf ("%d", &choice);

switch (choice)

{

case 1:

HEAD = insertbegin(HEAD);

break;

case 2:

HEAD = insertAtEnd(HEAD);

break;

case 3:

HEAD = insertAtPos(HEAD);

break;

default:

printf("\nError: Invalid input\n");

exit(0);

break;

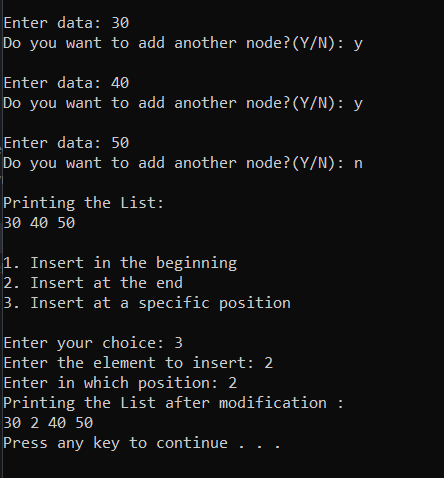
}

printf("Printing the List after modification : \n");

display(HEAD);

}

**Output:**



**Q5. WAP to Create a linked list and delete a new node as per the users choice.**

#include <stdio.h>

#include <stdlib.h>

struct Node{

int data;

struct Node\* next;

};

void display(struct Node\* ptr){

while(ptr != NULL){

printf("%d ", ptr->data);

ptr = ptr->next;

}

}

struct Node\* deletestart(struct Node \*HEAD)

{

struct Node \*p = HEAD;

p = p->next;

free(HEAD);

return p;

}

struct Node\* deleteAtPos(struct Node\* HEAD){

int pos;

printf("Enter the position: ");

scanf("%d", &pos);

struct Node\* ptr = HEAD;

int i=0;

while(i!=pos-2){

ptr = ptr->next;

i++;

}

struct Node\* p = ptr->next;

ptr->next = p->next;

free(p);

return HEAD;

}

struct Node\* deleteEnd(struct Node\* head){

struct Node\* ptr = head;

struct Node\* temp;

ptr = head;

while ((ptr->next)->next != NULL)

{

ptr = ptr->next;

}

temp = ptr->next;

ptr->next = NULL;

free(temp);

return head;

}

int main()

{

struct Node \*HEAD;

struct Node \*NEW;

struct Node \*PTR;

char choice;

HEAD = NULL;

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

HEAD = NEW;

PTR = HEAD;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

while (choice == 'Y' || choice == 'y')

{

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

PTR->next = NEW;

PTR = PTR->next;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

}

PTR = HEAD;

printf("\nPrinting the List: \n");

while (PTR != NULL)

{

printf("%d ", PTR->data);

PTR = PTR->next;

}

printf("\n\n");

printf ("1. Delete the first element\n");

printf ("2. Delete the last element\n");

printf ("3. Delete the node at a specific position\n");

printf ("\nEnter your choice: ");

scanf ("%d", &choice);

switch (choice)

{

case 1:

HEAD = deletestart(HEAD);

break;

case 2:

HEAD = deleteEnd(HEAD);

break;

case 3:

HEAD = deleteAtPos(HEAD);

break;

default:

printf("\nError: Invalid input\n");

exit(0);

break;

}

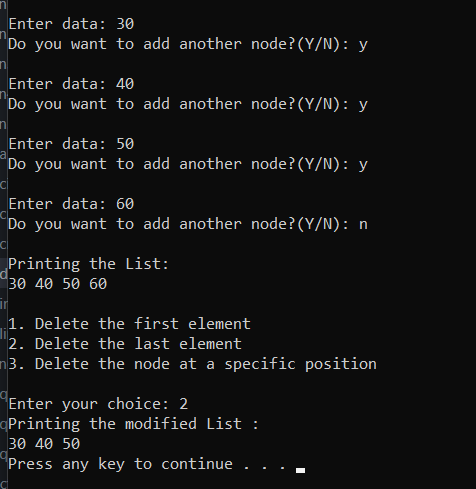
printf("Printing the modified List : \n");

display(HEAD);

return 0;

}

**Output:**



**Q6.  WAP to create a circular linked list and print all the elements.**

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;

};

int main(void)

{

struct Node \*HEAD;

struct Node \*NEW;

struct Node \*PTR;

char choice;

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = HEAD;

HEAD = NEW;

PTR = HEAD;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

while (choice == 'Y' || choice == 'y')

{

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = HEAD;

PTR->next = NEW;

PTR = PTR->next;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

}

PTR = HEAD;

printf("\nPrinting the List: \n");

do{

printf("%d ", PTR->data);

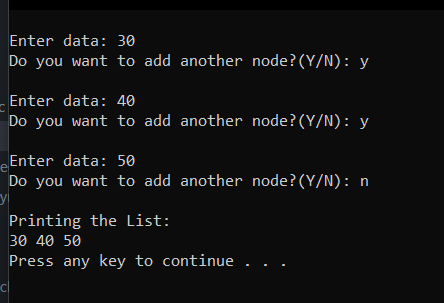
PTR = PTR->next;

}while (PTR != HEAD);

return 0;

}

**Output:**



**Q7. WAP to create a linked list and perform linear search.**

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node \*next;

};

void display(struct Node\* ptr)

{

printf("\nPrinting the List: \n");

while (ptr != NULL)

{

printf("%d ", ptr->data);

ptr = ptr->next;

}

}

int LinearSearch(struct Node\* head, int item)

{

struct Node\* ptr = head;

int pos = 1;

while (ptr != NULL)

{

if (ptr->data == item){

return pos;

}

ptr = ptr->next;

pos++;

}

return -1;

}

int main(void)

{

struct Node \*HEAD;

struct Node \*NEW;

struct Node \*PTR;

char choice;

HEAD = NULL;

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

HEAD = NEW;

PTR = HEAD;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

while (choice == 'Y' || choice == 'y')

{

NEW = (struct Node \*)malloc(sizeof(struct Node));

printf("\nEnter data: ");

scanf("%d", &NEW->data);

fflush(stdin);

NEW->next = NULL;

PTR->next = NEW;

PTR = PTR->next;

printf("Do you want to add another node?(Y/N): ");

scanf("%c", &choice);

}

display(HEAD);

int x;

printf ("\n\nEnter the element to search: ");

scanf ("%d", &x);

x = LinearSearch(HEAD, x);

if (x!=-1)

printf ("Element is in Position: %d\n", x);

else

printf ("Element not found\n");

return 0;

}

**Output:**

