1.wap to implement singly linked list.

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<conio.h>
4. struct node{
5. int a;
6. struct node \*next;
7. }\*start=NULL;
8. void creat(){
9. char ch;
10. do{
11. struct node \*new\_,\*ptr;
12. new\_=(struct node\*)malloc(sizeof(struct node));
13. printf("\nEnter data\n");
14. scanf("%d",&new\_->a);
15. new\_->next=NULL;
16. if(start == NULL){
17. start=new\_;
18. ptr=new\_;
19. }
20. else{
21. ptr->next=new\_;
22. ptr=new\_;
23. }
24. printf("Do you want to create a new node..... answer in y or n\n ");
25. ch=getche();
26. }while(ch == 'y');
27. }
28. void display()
29. {
30. struct node \*ptr;
31. printf("The Linked List is \n");
32. ptr=start;
33. while(ptr!=NULL)
34. {
35. printf("%d\n",ptr->a);
36. ptr=ptr->next;
37. }
38. }
39. void main(){
40. creat();
41. display();
42. getch();
43. }

Output :-

Enter data

5

Do you want to create a new node..... answer in y or n

y

Enter data

4

Do you want to create a new node..... answer in y or n

nThe Linked List is

5

4

2. program should have the facility to insert the element at the beginning ,at the end or in the middle of the list. Consider the same situation for deletion case also.

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct node{

int a;

struct node \*next;

}\*start=NULL;

void creat(){

char ch;

do{

struct node \*new\_,\*ptr;

new\_=(struct node\*)malloc(sizeof(struct node));

printf("\nEnter data\n");

scanf("%d",&new\_->a);

new\_->next=NULL;

if(start == NULL){

start=new\_;

ptr=new\_;

}

else{

ptr->next=new\_;

ptr=new\_;

}

printf("Do you want to create a new node..... answer in y or n\n ");

ch=getche();

}while(ch == 'y');}

void display()

{

struct node \*ptr;

printf("The Linked List is \n");

ptr=start;

while(ptr!=NULL)

{

printf("%d\n",ptr->a);

ptr=ptr->next;

}}

void insert\_beg(){

struct node \*ptr,\*new\_;

new\_=(struct node \*)malloc(sizeof(struct node));

printf("Enter data in node\n");

scanf("%d",&new\_->a);

if(start == NULL){

start=new\_;

ptr=new\_;

}

else{

new\_->next=start;

start=new\_;}}

void insert\_end(){

struct node \*ptr,\*new\_;

new\_=(struct node \*)malloc(sizeof(struct node));

printf("Enter data\n");

scanf("%d",&new\_->a);

new\_->next=NULL;

ptr=start;

while(ptr->next != NULL){

ptr=ptr->next;}

ptr->next=new\_;

}

void insert\_mid(){

struct node \*ptr,\*ptr1,\*new\_;

int i,pos;

ptr=start;

ptr1=start;

printf("Enter the position where you want to insert node\n");

scanf("%d",&pos);

new\_=(struct node \*)malloc(sizeof(struct node));

printf("Enter data\n");

scanf("%d",&new\_->a);

for(i=1;i<pos-1;i++){

ptr=ptr->next;

}

ptr1=ptr->next->next;

ptr->next=new\_;

new\_->next=ptr1;

}

void main(){

creat();

display();

insert\_beg();

printf("Insert at beginning\n");

display();

insert\_end();

printf("Insert at end\n");

display();

insert\_mid();

printf("Insert at mid\n");

display();

getch();

}

Output:-Enter data:5

Do you want to create a new node..... answer in y or n

y

Enter data: 45

Do you want to create a new node..... answer in y or n

The Linked List is

5

45

Enter data in node

5

Insert at beginning

The Linked List is

5

5

45

Enter data

6

Insert at end

The Linked List is

5

5

45

6

**3. wap to implement linked list in ascending order of the elements. Apply the four basic opertions on it**.

#include<stdio.h>

#include<stdlib.h>

struct node{

int a;

struct node \*next;

}\*start=NULL;

void create(){

char ch;

do{

struct node \*new\_,\*ptr;

new\_=(struct node\*)malloc(sizeof(struct node));

printf("\nEnter data\n");

scanf("%d",&new\_->a);

new\_->next=NULL;

if(start == NULL){

start=new\_;

ptr=new\_;

}

else{

ptr->next=new\_;

ptr=new\_;

}

printf("Do you want to create a new node..... answer in y or n\n ");

ch=getche();

}while(ch == 'y');

}

int count(){

int i,n=0;

struct node \*temp;

temp=start;

for(i=1;temp!=NULL;i++){

n++;

temp=temp->next;

}

return n;

}

void ascend(int x){

struct node \*p,\*q;

int i,j,k,temp;

k=x;

for(i=1 ; i<x ;i++,k--){

p=start;

q=p->next;

for(j=1 ; j<k ; j++){

if(p->a > q->a){

temp=p->a;

p->a=q->a;

q->a=temp;

}

p=p->next;

q=q->next;

}

}

}

void display()

{

struct node \*ptr;

printf("The Linked List is \n");

ptr=start;

while(ptr!=NULL)

{

printf("%d\n",ptr->a);

ptr=ptr->next;

}

}

int main(){

int s;

create();

display();

s=count();

printf("no of data is %d\n",s);

ascend(s);

display();

return 0;

}

**Output:-**

**Enter data:4**

**Do you want to create a new node..... answer in y or n**

**y**

**Enter data**

**9**

**Do you want to create a new node..... answer in y or n**

**y**

**Enter data:3**

**Do you want to create a new node..... answer in y or n**

**nThe Linked List is**

**4**

**9**

**3**

**no of data is 3**

**The Linked List is:3**

**4**

**9**

**7.wap to implement doubly linked list.**

**#include<stdio.h>**

**#include<malloc.h>**

**#include<conio.h>**

**typedef struct Node**

**{**

**struct Node \*prev;**

**int info ;**

**struct Node \*next;**

**}node;**

**void createdub(node\*\*,node\*\*);**

**void insertAtBeg(node \*\*,node\*\*,int);**

**void display(node \*);**

**void main()**

**{**

**int ch, item, pos;**

**node \*start ,\*end ;**

**start = end = NULL;**

**createdub(&start,&end);**

**printf("\nThe list is : ");**

**display(start);**

**printf("\n\nEnter the item to be inserted at beginning :");**

**scanf("%d",&item);**

**insertAtBeg(&start,&end,item);**

**printf("\nNow the list is : ");**

**display(start);**

**getch();**

**}**

**void createdub(node \*\*start,node \*\*end)**

**{**

**int i,item ,k=1;**

**printf("\nEnter number of node: ");**

**scanf("%d",&i);**

**while(i)**

**{**

**node \*ptr;**

**printf("\nEnter the info for node %d : ",k);**

**scanf("%d",&item);**

**ptr=(node\*)malloc(sizeof(node));**

**ptr->info=item;**

**if(\*start==NULL)**

**{**

**ptr->prev = ptr->next = NULL ;**

**\*start = \*end = ptr ;**

**}**

**else**

**{**

**ptr->prev = \*end;**

**(\*end)->next = ptr ;**

**ptr->next= NULL;**

**(\*end)=ptr;**

**}**

**i--;**

**k++;**

**}**

**}**

**void insertAtBeg(node \*\*start,node \*\*end,int item )**

**{**

**node \*ptr;**

**ptr=(node\*)malloc(sizeof(node));**

**ptr->info=item;**

**if(\*start==NULL)**

**{**

**\*start=ptr;**

**\*end=ptr;**

**}**

**else**

**{**

**ptr->prev = NULL;**

**ptr->next=\*start;**

**(\*start)->prev=ptr;**

**\*start=ptr;**

**}**

**}**

**void display(node \*start)**

**{**

**while(start !=NULL)**

**{**

**printf("\t %d",start->info);**

**start = start->next;**

**}**

**Output:-**

**Enter number of node: 3**

**Enter the info for node 1 : 25**

**Enter the info for node 2 : 36**

**Enter the info for node 3 : 45**

**The list is : 25 36 45**

**Enter the item to be inserted at beginning :36**

**Now the list is : 36 25 36 45**

**4. wap to join two list.**

**#include <stdio.h>**

**struct node**

**{**

**int data;**

**struct node \*next;**

**};**

**void push(struct node \*\* head\_ref, int new\_data){**

**struct node\* new\_node =**

**(struct node\*) malloc(sizeof(struct node));**

**new\_node->data = new\_data;**

**new\_node->next = (\*head\_ref);**

**(\*head\_ref) = new\_node;}**

**void printList(struct node \*head)**

**{**

**struct node \*temp = head;**

**while (temp != NULL)**

**{**

**printf("%d ", temp->data);**

**temp = temp->next;**

**}**

**printf("\n");**

**}**

**void merge(struct node \*p, struct node \*\*q)**

**{**

**struct node \*p\_curr = p, \*q\_curr = \*q;**

**struct node \*p\_next, \*q\_next;**

**while (p\_curr != NULL && q\_curr != NULL)**

**{**

**p\_next = p\_curr->next;**

**q\_next = q\_curr->next;**

**q\_curr->next = p\_next; // Change next pointer of q\_curr**

**p\_curr->next = q\_curr; // Change next pointer of p\_curr**

**p\_curr = p\_next;**

**q\_curr = q\_next;**

**}**

**\*q = q\_curr; // Update head pointer of second list**

**}**

**int main()**

**{**

**struct node \*p = NULL, \*q = NULL;**

**push(&p, 3);**

**push(&p, 2);**

**push(&p, 1);**

**printf("First Linked List:\n");**

**printList(p);**

**push(&q, 8);**

**push(&q, 7);**

**push(&q, 6);**

**push(&q, 5);**

**push(&q, 4);**

**printf("Second Linked List:\n");**

**printList(q);**

**merge(p, &q);**

**printf("Modified First Linked List:\n");**

**printList(p);**

**printf("Modified Second Linked List:\n");**

**printList(q);**

**getchar();**

**return 0;}**

**output:**

**First Linked List:1 2 3**

**Second Linked List:**

**4 5 6 7 8**

**Modified First Linked List:**

**1 4 2 5 3 6**

**Modified Second Linked List:**

**7 8**

**5.wap to implement a singly linked list.delete any node of the list. Append it to the beginning of the list.**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct node{**

**int a;**

**struct node \*next;**

**}\*start=NULL;**

**void create(){**

**struct node \*new\_,\*ptr;**

**char ch='y';**

**while(ch == 'y' || ch == 'Y'){**

**new\_=(struct node\*)malloc(sizeof(struct node));**

**new\_->next=NULL;**

**printf("\nEnter the data\n");**

**scanf("%d",&new\_->a);**

**new\_->next=NULL;**

**if(start == NULL){**

**start=new\_;**

**ptr=new\_; }**

**else{**

**ptr->next=new\_;**

**ptr=new\_; }**

**printf("Do you want to create new node y or no ?\n");**

**ch=getche();}}**

**void apend(struct node \*t){**

**t->next=start;**

**start=t;}**

**void display(){**

**struct node \*ptr;**

**ptr=start;**

**while(ptr!=NULL){**

**printf("\n%d\n",ptr->a);**

**ptr=ptr->next;}}**

**void deletenode(){**

**int i,pos;**

**struct node \*ptr,\*temp,\*temp1;**

**ptr=start;**

**printf("Enter the position of node which you want to delete\n");**

**scanf("%d",&pos);**

**for(i=1;i<pos-1;i++){**

**ptr=ptr->next;}**

**temp=ptr;**

**temp1=temp->next->next;**

**ptr=ptr->next;**

**apend(ptr);**

**temp->next=temp1;}**

**int main(){**

**create();**

**display();**

**deletenode();**

**printf("Linked list after delete and apend\n");**

**display();**

**return 0;**

**}**

**Output:**

**Enter the data**

**6**

**Do you want to create new node y or no ?**

**y**

**Enter the data3**

**Do you want to create new node y or no ?**

**N**

**6**

**3**

**Enter the position of node which you want to delete**

**3**

**Linked list : 6**

**8.wap to delete the nodes whose**