#include <stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*next;

};

struct node \*start=NULL;

void insert\_begin()

{

struct node \*temp;

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

if(temp==NULL)

{

printf("\nOut of Memory Space:\n");

return;

}

printf("\nEnter the data value for the node:\t" );

scanf("%d",&temp->info);

temp->next =NULL;

if(start==NULL)

{

start=temp;

}

else

{

temp->next=start;

start=temp;

}

}

void insert\_end()

{

struct node \*temp,\*ptr;

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

if(temp==NULL)

{

printf("\nOut of Memory Space:\n");

return;

}

printf("\nEnter the data value for the node:\t" );

scanf("%d",&temp->info );

temp->next =NULL;

if(start==NULL)

{

start=temp;

}

else

{

ptr=start;

while(ptr->next !=NULL)

{

ptr=ptr->next ;

}

ptr->next =temp;

}

}

void insert\_pos()

{

struct node \*ptr,\*temp;

int i,pos;

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

if(temp==NULL)

{

printf("\nOut of Memory Space:\n");

return;

}

printf("\nEnter the position for the new node to be inserted:\t");

scanf("%d",&pos);

printf("\nEnter the data value of the node:\t");

scanf("%d",&temp->info) ;

temp->next=NULL;

if(pos==0)

{

temp->next=start;

start=temp;

}

else

{

for(i=0,ptr=start;i<pos-1;i++)

{

ptr=ptr->next;

if(ptr==NULL)

{

printf("\nPosition not found:[Handle with care]\n");

return;

}

}

temp->next =ptr->next ;

ptr->next=temp;

}

}

void traversal()

{

struct node \*ptr;

if(start==NULL)

{

printf("\nList is empty:\n");

return;

}

else

{

ptr=start;

printf("\nThe List elements are:\n");

while(ptr!=NULL)

{

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

ptr=ptr->next ;

}

}

printf("\n");

}

int main()

{

int ch;

while(1){

printf("---------------------------------------\n");

printf("1.Insertion at beginning\t2.Insertion at end\t3.Insertion at specific position\t4.Traversal\t5.Exit");

printf("\nEnter your choice:\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

insert\_begin();

break;

case 2:

insert\_end();

break;

case 3:

insert\_pos();

break;

case 4:

traversal();

break;

case 5:

printf("\nExiting....\n");

exit(0);

default:

printf("\nInvalid choice!!!\n");

return;

}

}

return 0;

}