#include<stdio.h>

#include<math.h>

#include<stdlib.h>

#include<malloc.h>

typedef struct link//to shorten name

        {

            int info;

            struct link \*next;

     struct link \*prev;

        }node;

node \* create(int n);

node \* search(node \*p,int n);

node \* insert(node \*p,int n,int position,int \*num);

void print(node \* p);

void modify(node \*p,int n1,int n2);

node \* del(int ele,node \* p);

int main()

{

    int i,ch,n,ele,n1,n2,num;

    node \*start;node \*p;

    node \*pos;

    char justforfun;

int \*a;//to store address of no. of elements

printf("\nCreate your List (^\_^)\n");

printf("\nEnter the size of the Linked List\n");

    scanf("%d",&num);

    start=create(num);

    a=&num;

    printf("\nLinked List successfully created...\n");

    while(1)//so that programme runs any no. of times

    {

        printf("\n\nPress 1 to create your linked list\n");//menu

        printf("Press 2 to search an element in the list\n");

        printf("Press 3 to insert an element in your list\n");

        printf("Press 4 to delete an entry\n");

        printf("Press 5 to print your linked list\n");

        printf("Press 6 to modify a value\n");

        printf("Press 7 to exit from the programme\n");

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

        scanf("%d",&ch);

        switch(ch)

        {

            case 1:printf("\nEnter the size of the Linked List --> ");

                   scanf("%d",&num);

                   start=create(num);

       a=&num;

       printf("Linked List successfully created...\n");

                   break;

            case 2:printf("Enter the element to be searched --> ");

                   scanf("%d",&ele);

                   pos=search(start,ele);

                   if (pos==NULL)

                    printf("\nElement does not exist\n");

                   else

                    printf("\nElement Found\n");

                   break;

            case 3:printf("Enter the element to be inserted --> ");

                   scanf("%d",&ele);

                   printf("Enter the position where the element is to be inserted --> ");

                   scanf("%d",&n);

   if((n>num+1))

      printf("List has positions upto %d only\n",num+1);

   else

                      start=insert(start,ele,n,a);

       printf("Element inserted in the Linked List...\n");

                   break;

            case 4:printf("Enter the value to be deleted --> ");

                   scanf("%d",&n);

                   start=del(n,start);

       printf("Element deleted...\n");

                   break;

            case 5:print(start);

                   break;

            case 6:printf("Enter the value to be modified --> ");

                   scanf("%d",&n1);

                   printf("Enter the value to be kept --> ");

                   scanf("%d",&n2);

                   modify(start,n1,n2);

       printf("Value modified...\n");

                   break;

            case 7: exit(0);//to end while loop

            default : printf("Invalid Choice\n");

        }

        printf("\n\n\nPress a key to continue");

        scanf("%c",&justforfun);

        printf("\n\n\n");

    }

    return 0;

}

node \* create(int n)

{

    node \*st=NULL;

    node \*p,\*l,\*q;

    int i,x;

    for(i=1;i<=n;i++)

    {

        p=(node \*)malloc(sizeof(node));//to allocate memory

        printf("Enter Data [%d] --> ",i);

        scanf("%d",&p->info);

        p->next=NULL;

        if(st==NULL)

        {

     st=p;

     st->prev=NULL;

        }

else

        {

     l->next=p;

p->prev=l;

}

        l=p;

    }

    return (st);

}

node \* search(node \*p,int n)

{

    while(p!=NULL && p->info!=n)

    {

        p=p->next;

    }

    return p;

}

void print(node \* p)

{

    printf("\nList --> \n");

    while(p!=NULL)

    {

        int i;

        i=p->info;

        printf("%d ",i);

        p=p->next;

    }

}

node \* insert(node \* p,int ele,int n,int \*num)

{

    node \*st;

    node \*q;

node \*l;

node \*w;

    int i;

q=p;

l=p;

if(n==1)

{

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

st->info=ele;

st->next=p;

st->prev=NULL;

\*num=\*num+1;

return st;

}

else

{

     for(i=1;i<n;i++)

     {

q=p;

         p=p->next;

     }

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

st->info=ele;

st->prev=q;

st->next=p;

q->next=st;

\*num=\*num+1;

return l;

}

}

void modify(node \*p,int n1,int n2)

{

    while(p->info!=n1 && p!=NULL)

    {

        p=p->next;

    }

    if(p==NULL)

        printf("Element to be changed was not found\n");

    else

    {

        p->info=n2;

    }

}

node \* del(int n,node \*p)

{

    node \*q,\*r,\*start;

    start=p;

    while((p!=NULL) && (p->info!=n))

    {

        q=p;

        p=p->next;

    }

    if(p==NULL)

    {

            printf("Element not Found\n");

            return start;

    }

    else

    {

        if(p==start)

        {

start=start->next;

start->prev=NULL;

}

        else

        {

            r=p->next;

            q->next=r;

r->prev=q;

        }

        free(p);//to avoid dangling pointers

        return start;

    }

}