#include<stdio.h>

typedef struct node

{

int data;

struct node \*next;

}node;

node\* create();

void print(node\*);

node\* ifn(node\*);

node\* iln(node\*);

node\* ibn(node\*);

node\* dfn(node\*);

node\* dln(node\*);

node\* dbn(node\*);

node\* updatenode(node\*);

int searchnode(node\*);

int countnode(node\*);

void main( )

{

node \*HEAD=NULL;

int choice,res;

while(1)

{

printf("\n=====================\n");

printf("\n1 : Create SLL.");

printf("\n2 : Print SLL.");

printf("\n3 : Insert Node at First in SLL.");

printf("\n4 : Insert Node at Last in SLL.");

printf("\n5 : Insert Node between in SLL.");

printf("\n6 : Delete Node First from SLL.");

printf("\n7 : Delete Node Last from SLL.");

printf("\n8 : Delete Node between SLL.");

printf("\n9 : Search Node in SLL.");

printf("\n10: Update Node in SLL.");

printf("\n11: Count Nodes in SLL.");

printf("\n12: EXIT.");

printf("\n=====================\n");

printf("\nEnter Your Choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1:

HEAD=create();

printf("\nSuccess Create SLL!!!\n");

break;

case 2:

print(HEAD);

break;

case 3:

HEAD=ifn(HEAD);

printf("\nSuccess Inserted First Node in SLL!!!\n");

break;

case 4:

HEAD=iln(HEAD);

printf("\nSuccess Inserted Last Node in SLL!!!\n");

break;

case 5:

HEAD=ibn(HEAD);

printf("\nSuccess Inserted Between Node in SLL!!!\n");

break;

case 6:

HEAD=dfn(HEAD);

printf("\nSuccess Deleted First Node from SLL!!!\n");

break;

case 7:

HEAD=dln(HEAD);

printf("\nSuccess Deleted Last Node from SLL!!!\n");

break;

case 8:

HEAD=dbn(HEAD);

printf("\nSuccess Deleted Between Node from SLL!!!\n");

break;

case 9:

res=searchnode(HEAD);

if(res!=-1)

printf("\nFound node at %d!!!\n",res);

else

printf("\nNot Found!!!\n");

break;

case 10:

HEAD=updatenode(HEAD);

printf("\nSuccess Updated Node in SLL!!!\n");

break;

case 11:

res=countNode(HEAD);

printf("\nNumber of nodes are %d!!!\n",res);

break;

case 12:

exit(0);

default:

printf("\nPlease Select Proper Choice[1-12]!!!");

}

}

}

node\* create()

{

int n,value,i=1;

node \*head,\*p,\*q;

printf("\nHow Many Nodes U Want : ");

scanf("%d",&n);

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

printf("\nEnter Data : ");

scanf("%d",&value);//10

head->data=value;

head->next=head;

p=head;

// head p

//|10-2002| --->|20-3002| --->|30-1002|

// 1002 2002 3002

while(i<n)

{

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

printf("\nEnter Data : ");

scanf("%d",&value);//20

q->data=value;

q->next=head;

p->next=q;

p=p->next;

i++;

}

return head;

}

void print(node \*head)

{

node \*p;

p=head;

printf("\n");

if(p!=NULL)

{

while(p->next!=head)//1002!=1002

{

printf("%d=>",p->data);

p=p->next;

}

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

}

else

{

printf("\nSLL Empty!!!");

}

}

node\* ifn(node \*head)

{

int value;

node \*p,\*q;

// p head

// |5-1002|--> |10-2002| --->|20-3002| --->|30-1002|

// 4002 1002 2002 3002

q=head;

while(q->next!=head)

q=q->next;

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

printf("\nEnter Data : ");

scanf("%d",&value);//10

p->data=value;

p->next=head;

head=p;

q->next=p;

return head;

}

node\* iln(node \*head)

{

int value;

node \*p,\*q;

p=head;

while(p->next!=head)

p=p->next;

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

printf("\nEnter Data : ");

scanf("%d",&value);//10

q->data=value;

q->next=head;

p->next=q;

return head;

}

node\* ibn(node \*head)

{

int value,i=1,pos;

node \*p,\*q;

p=head;

printf("\nEnter Position : ");

scanf("%d",&pos);//10

while(i<pos-1)

{

p=p->next;

i++;

}

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

printf("\nEnter Data : ");

scanf("%d",&value);//10

q->data=value;

q->next=p->next;

p->next=q;

return head;

}

node\* dfn(node \*head)

{

node \*p,\*q;

p=q=head;

while(q->next!=head)

q=q->next;

head=head->next;

free(p);

q->next=head;

return head;

}

node\* dln(node \*head)

{

node \*p,\*q;

p=head;

while(p->next->next!=head)

p=p->next;

q=p->next;

free(q);

p->next=head;

return head;

}

node\* dbn(node \*head)

{

int i=1,pos;

node \*p,\*q;

p=head;

printf("\nEnter Position : ");

scanf("%d",&pos);//10

while(i<pos-1)

{

p=p->next;

i++;

}

q=p->next;

p->next=q->next;

free(q);

return head;

}

int countNode(node \*head)

{

int cnt=0;

node \*p;

p=head;

if(p==NULL)

return 0;

while(p->next!=head)

{

cnt++;

p=p->next;

}

cnt++;

return cnt;

}

int searchnode(node \*head)

{

int value,i=0;

node \*p;

p=head;

printf("\nEnter Value u want Search : ");

scanf("%d",&value);//10

if(p==NULL)

{

printf("\nInvalid Operation Empty SLL");

return -1;

}

while(p->next!=head)

{

i++;

if(p->data==value)

return i;

p=p->next;

}

if(p->data==value)

return i;

return -1;

}

node\* updatenode(node \*head)

{

int ovalue,nvalue;

node \*p;

p=head;

if(p==NULL)

{

printf("\nInvalid Operation Empty SLL");

return head;

}

printf("\nEnter Old Value: ");

scanf("%d",&ovalue);//30

printf("\nEnter New Value: ");

scanf("%d",&nvalue);//35

//10 20 35 40

while(p!=NULL)

{

if(p->data==ovalue)

{

p->data=nvalue;

return head;

}

p=p->next;

}

return head;

}