#include<stdio.h>

#include<stdlib.h>

struct DNode

{

struct DNode \*left;

int data;

struct DNode \*right;

}\*first=NULL,\*temp,\*current;

int n,pos,item,ch,i;

typedef struct DNode DNode;

void insertDlistf();

void insertDlistl();

void insertDlistatany();

void deleteDlistf();

void deleteDlistl();

void deleteDlistatany();

void traverseDlist();

void main()

{int r,p;

printf("Enter the number of nodes:\n");

scanf("%d",&n);

while(n!=8)

{

printf("\n1.Insertf\n2.Insertl\n3.Insertan\n4.Deletef\n5.Deletel\n6.Deletean\n7.Traverse\n8.Exit");

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

scanf("%d",&ch);

switch(ch)

{

case 1:{

printf("Enter the item to be inserted: ");

scanf("%d",&r);

insertDlistf(&first,r);

break;

}

case 2:

{printf("Enter item:\n");

scanf("%d",&r);

insertDlistl(&first,r);

break;

}

case 3:

{

printf("Enter the item:\n");

scanf("%d",&r);

printf("Enter the position:\n");

scanf("%d",&p);

insertDlistatany(&first,p,r);

break;

}

case 4:

{

deleteDlistf(&first);

break;

}case 5:

{

deleteDlistl(&first);

break;

}

case 6:

{

printf("Enter the position to be deleted:\n");

scanf("%d",&p);

deleteDlistatany(&first,p);

break;

}

case 7:

{

traverseDlist(first);

break;

}

case 8:

{

printf("Exit\n");

break;

}

default:

{

printf("Enter a valid choice\n");

break;

}

}}

}

void insertDlistf( DNode \*\*first,int item)

{

temp=(DNode\*)malloc(sizeof(DNode));

temp->data=item;

temp->left=NULL;

if(\*first==NULL)

temp->right=NULL;

else

{

temp->right=(\*first);

(\*first)->left=temp;

}

(\*first)=temp;

}

void insertDlistl(DNode \*\*first,int pos,int item)

{

temp=(DNode\*)malloc(sizeof(DNode));

temp->data=item;

temp->right=NULL;

if(\*first==NULL)

{

temp->left=NULL;

(\*first)=temp;

}

else

{

current=(\*first);

while(current->right!=NULL)

current=current->right;

temp->left=current;

current->right=temp;

}

}

void insertDlistatany(DNode \*\*first,int pos,int item)

{

temp=(DNode\*)malloc(sizeof(DNode));

temp->data=item;

if(pos==1)

{

if(\*first!=NULL)

(\*first)->left=temp;

temp->right=(\*first);

temp->left=NULL;

(\*first)=temp;

}

else

{

i=2;

current=(\*first);

while((i<pos)&&(current->right!=NULL))

{

i++;

current=current->right;}

temp->left=current;

temp->right=current->right;

if(current->right!=NULL)

temp->right->left=temp;

current->right=temp;

}

}

void deleteDlistf(DNode \*\*first)

{

if(\*first==NULL)

{

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

}

current=(\*first);

(\*first)=(\*first)->right;

if(\*first!=NULL)

(\*first)->left=NULL;

item=current->data;

free(current);

printf("\nDeleted Item=%d",item);

}

void deleteDlistl(DNode \*\*first)

{

if(\*first==NULL)

{

printf("List is empty\n");

return;

}

current=\*first;

while(current->right!=NULL)

current=current->right;

if(current->left!=NULL)

current->left->right=current->right;

else

(\*first)=NULL;

item=current->data;

free(current);

printf("Deleted item= %d",item);

}

void deleteDlistatany(DNode \*\*first,int pos)

{i=1;

if(\*first==NULL)

{

printf("List is empty\n");

return;

}

current=(\*first);

while(current!=NULL)

{

if(i==pos)

{

item=current->data;

if(current->left==NULL)

{

current->right->left=NULL;

(\*first)=current->right;

free(current);

}

else if(current->right==NULL)

{

current->left->right=current->right;

free(current);

}

else

{

current->left->right=current->right;

current->right->left=current->left;

free(current);

}

printf("Deleted item=%d",item);

return;

}

i++;

current=current->right;}

printf("Such a node doesn't exist");

}

void traverseDlist(DNode \*first)

{

if(first==NULL)

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

else

{

while(first!=NULL)

{

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

first=first->right;

}

printf(" Last\n");}}































