#include <stdio.h>

#include <cs50.h>

#include <string.h>

#include <ctype.h>

//создали структуру dllnode включающую в себя

// val типа int

// и указатели prev и next

typedef struct dllist

{

int val;

struct dllist\* prev;

struct dllist\* next;

}

dllnode;

// прототипы функций

dllnode\* create(int value);

bool linsearch(dllnode\* head, int value);

void destroy(dllnode\* head);

dllnode\* insert(dllnode\* head, int value);

void showlist (dllnode\* head);

dllnode\* delelement(dllnode\* head, int number);

int main (void)

{

dllnode\* new = create(19);

new = insert(new, 11);

new = insert(new, 7);

new = insert(new, 9);

new = insert(new, -13);

new = insert(new, 144);

showlist(new);

new = delelement(new, 0);

showlist(new);

new = delelement(new, 4);

showlist(new);

new = delelement(new, 1);

showlist(new);

destroy(new);

}

// удаляет элемент под номером number

dllnode\* delelement(dllnode\* head, int number)

{

// ввели копию указатели и отвели его на элемент number

dllnode\* ptrcurr = head;

for (int i = 0; i < number; i ++)

{

ptrcurr = ptrcurr->next;

}

// проверка на 1 элемент списка

if (number == 0)

{

dllnode\* ptr = head;

ptr = head->next;

free(head);

ptr->prev = NULL;

return ptr;

}

// проверка на последний элемент списка

else if (ptrcurr->next == NULL)

{

dllnode\* ptrprev = head;

for (int i = 0; i < number-1; i ++)

{

ptrprev = ptrprev->next;

}

ptrprev->next = ptrcurr->next;

free(ptrcurr);

return head;

}

else

{

dllnode\* ptrprev = head;

dllnode\* ptrnext = head;

for (int i = 0; i < number-1; i ++)

{

ptrprev = ptrprev->next;

}

for (int i = 0; i < number+1; i ++)

{

ptrnext = ptrnext->next;

}

ptrprev->next = ptrcurr->next;

ptrnext->prev = ptrcurr->prev;

free(ptrcurr);

return head;

}

}

// выводит весь связанный список

void showlist (dllnode\* head)

{

if (head != NULL)

{

printf("%i ", head->val);

showlist(head->next);

}

else

{

printf("\n");

}

}

// вставляет элемент в начало списка

dllnode\* insert(dllnode\* head, int value)

{

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

if (ptr == NULL)

{

return NULL;

}

ptr->val = value;

ptr->next = head;

ptr->prev = NULL;

head->prev = ptr;

return ptr;

}

// удаляем список типа dllnode

void destroy(dllnode\* head)

{

if (head->next != NULL)

{

destroy(head->next);

head->next = NULL;

head->prev = NULL;

}

if (head->next == NULL)

{

free(head);

}

}

//функция ищет в dllnode значение value

bool linsearch(dllnode\* head, int value)

{

dllnode\* ptr = head;

if (ptr->val == value)

{

return true;

}

while (ptr->next != NULL)

{

if (ptr->val == value)

{

return true;

}

else

{

ptr = ptr->next;

}

}

return false;

}

// создаём переменную типа dllnode

dllnode\* create(int value)

{

//выделили память

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

//проверили на ошибки

if (ptr == NULL)

{

return NULL;

}

ptr->val = value;

ptr->prev = NULL;

ptr->next = NULL;

return ptr;

}