1. Створити однозв’язний лінійний список з 5-ти елементів, які мають одне інформаційне поле цілого типу (як додатні, так і від’ємні значення). На основі створеного списку створити ще 2 списки: в одному повинні міститися додатні елементи початкового списку, в іншому – від’ємні. Вивільнити динамічну пам’ять.

#include <stdio.h>

#include <stdlib.h>

#define NUMELEM 5

typedef struct mylist

{ int val;

struct mylist \*next;

}mylist;

mylist \*new\_node(int v)

{

mylist \*new;

if((new=(mylist\*)malloc(sizeof(mylist)))==NULL)return NULL;

new->val=v;

new->next=NULL;

return new;

}

mylist \*add\_front(mylist \*p, mylist \*n)

{

n->next=p;

return n;

}

void print\_mylist(mylist \*l)

{

for(; l!=NULL; l=l->next)

{

printf("%d ", l->val);

}

}

void free\_mylist(mylist \*l)

{

mylist \* n;

for(; l!=NULL; l=n)

{

n=l->next;

free(l);

}

}

int main(void)

{

mylist\* a=NULL, \*b=NULL, \*f=NULL;

int e=0, c=0;

for(int i=1; i<=NUMELEM; ++i)

{

scanf("%d", &e);

a=add\_front(a, new\_node(e));

if(e<0)b=add\_front(b, new\_node(e));

if(e>0)f=add\_front(f, new\_node(e));

}

print\_mylist(a);

puts("");

if(b!=NULL)print\_mylist(b);

puts("");

if(f!=NULL)print\_mylist(f);

free\_mylist(a);

free\_mylist(b);

free\_mylist(f);

return 0;

}