

```

//Количество элементов, больших своих соседей (традиционных локальных максимумов)
//_____ loc.h заголовочный файл
#include<stdio.h>
int number_loc (FILE *);
//_____ loc.c файл с реализациями
#include"loc.h"//включение заголовочного
inint number_loc(FILE *in)
{
    int n=0,s=0;//n – счетчик числа элементов, s – количество лок. макс
    ccur,prev=0,pprev; //текущий, предыдущий и предпредыдущий элементы
    //последовательности
    if(in==0)
    {
        return -1; //файл не открылся
    }
    while(fscanf(in,"%d",&cur)==1)//пока в последовательности есть текущий элемент
    {
        n++;
        if(n>2 && prev>pprev && prev>cur || n==2 && cur<prev)//если предыдущий явл.
        //локальным максимумом
            s++;
        pprev=prev; //меняем предпред. На предыдущий
        prev=cur; //меняем предыдущий на текущий
    }
    if(!feof(in)) // если конец файла не достигнут
    {
        return -2;
    }
    if(n==0)
    {
        return -3; // файл пустой
    }
    if (cur>pprev)// если последний эл-т явл. Лок. Макс. (предпоследний сохранился в pprev!!!!
        s++;

```

```

        return s;
    }
//_____ файл main.cpp
#include "loc.h" //включение заголовочного
int main(void)
{
    FILE *in=fopen("1.txt","r");
    int n;
    n=number_loc(in);
    switch(n)
    {
        case -1:
            printf("File not found\n");
            return -1;
        case -2:
            printf("File is bad\n");
            break;
        case -3:
            printf("File is empty");
            break;
        default:
            printf("number=%d\n",n);
    }
    fclose(in);
    return 0;
}

```

Компилируем только файлы, содержащие коды программ, заголовочные сюда не относятся!
gcc main.c loc.c
запускаем как раньше
./a.out