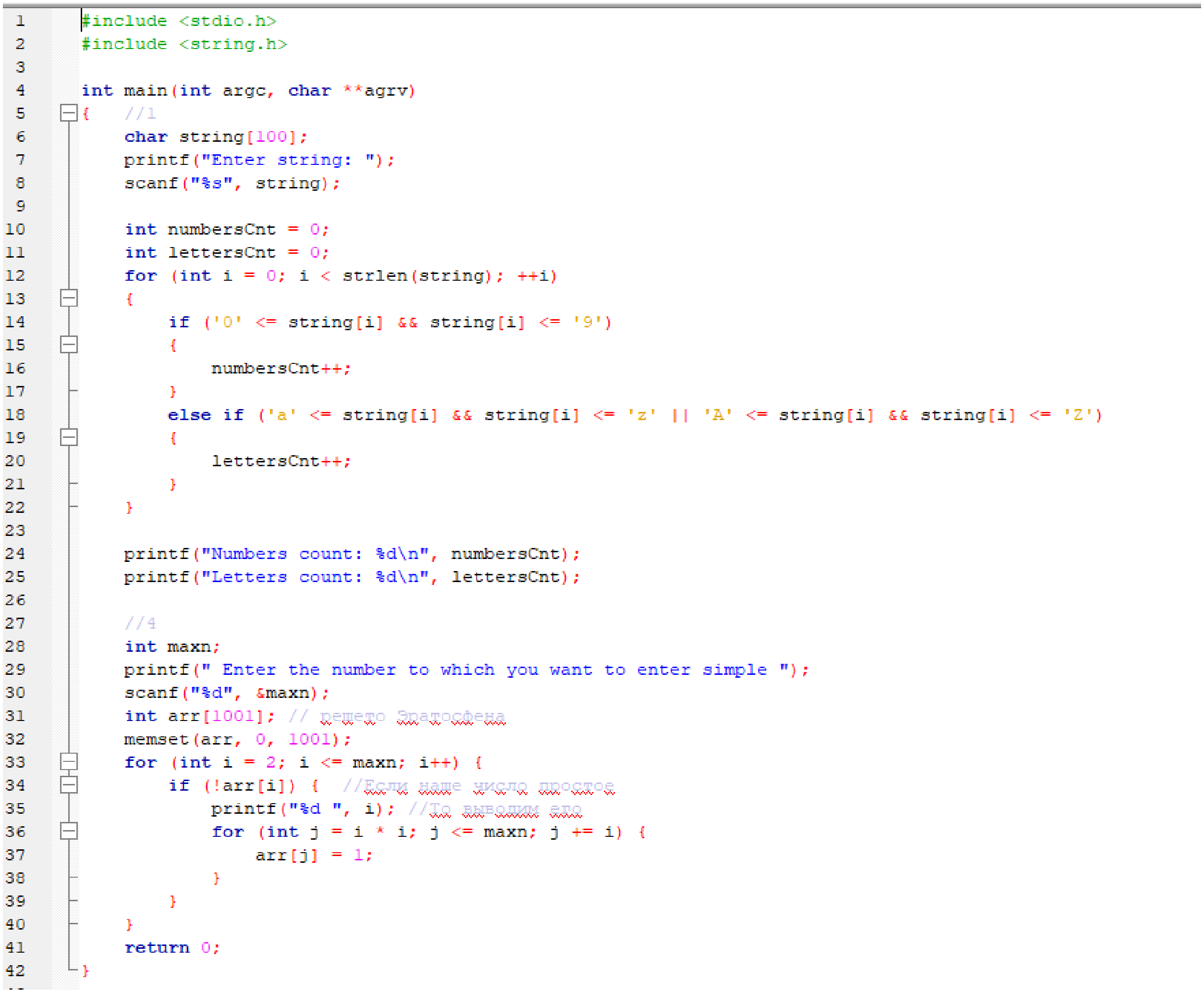
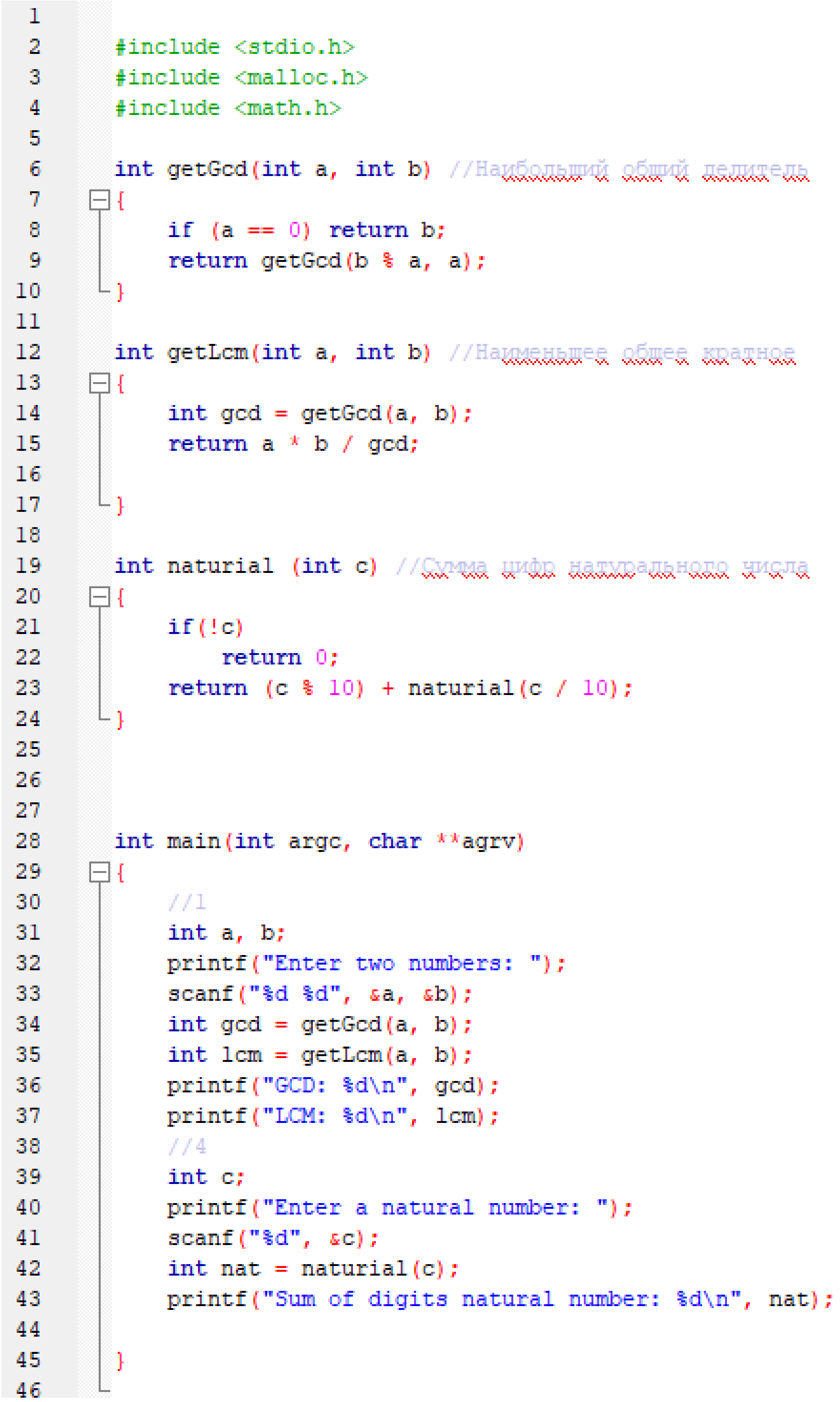
9laba.



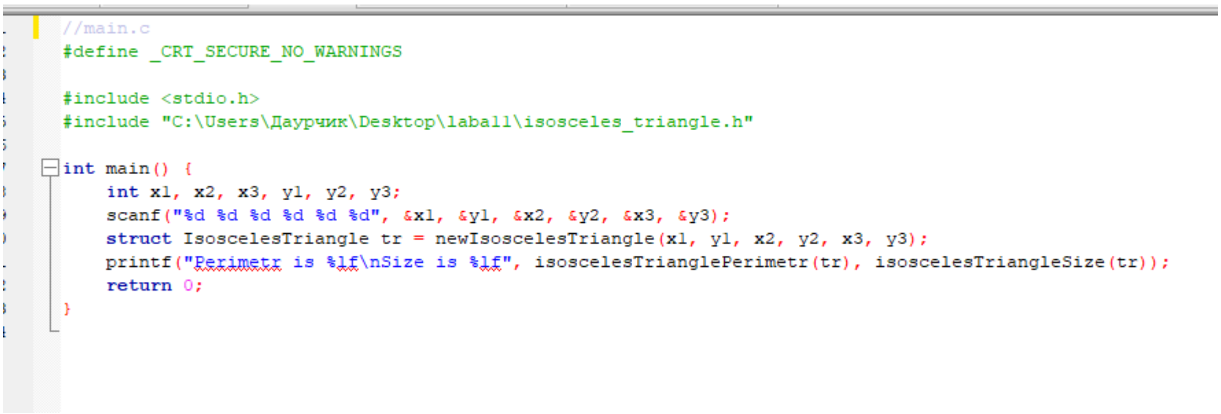
```c
#include <stdio.h>
#include <string.h>

int main(int argc, char **agrv)
{   //1
    char string[100];
    printf("Enter string: ");
    scanf("%s", string);

    int numbersCnt = 0;
    int lettersCnt = 0;
    for (int i = 0; i < strlen(string); ++i)
    {
        if ('0' <= string[i] && string[i] <= '9')
        {
            numbersCnt++;
        }
        else if ('a' <= string[i] && string[i] <= 'z' || 'A' <= string[i] && string[i] <= 'Z')
        {
            lettersCnt++;
        }
    }

    printf("Numbers count: %d\n", numbersCnt);
    printf("Letters count: %d\n", lettersCnt);

    //4
    int maxn;
    printf(" Enter the number to which you want to enter simple ");
    scanf("%d", &maxn);
    int arr[1001]; // решето Эратосфена
    memset(arr, 0, 1001);
    for (int i = 2; i <= maxn; i++) {
        if (!arr[i]) {  //Если наше число простое
            printf("%d ", i); //То выводим его
            for (int j = i * i; j <= maxn; j += i) {
                arr[j] = 1;
            }
        }
    }
    return 0;
}
```

10laba.



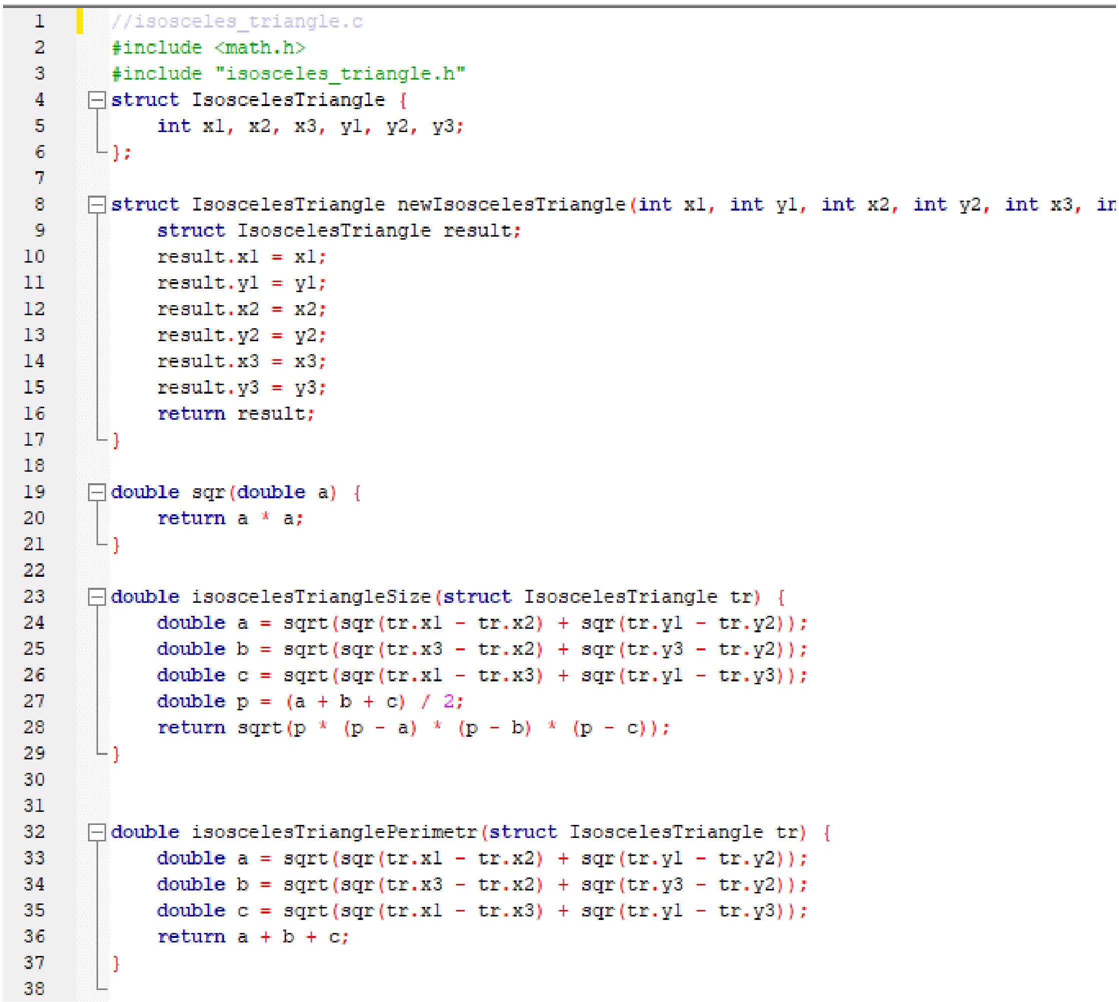
```

#include <stdio.h>
#include <malloc.h>
#include <math.h>

int getGcd(int a, int b) //Наибольший общий делитель
{
    if (a == 0) return b;
    return getGcd(b % a, a);
}

int getLcm(int a, int b) //Наименьшее общее кратное
{
    int gcd = getGcd(a, b);
    return a * b / gcd;

}

int naturial (int c) //Сумма цифр натурального числа
{
    if(!c)
        return 0;
    return (c % 10) + naturial(c / 10);
}



int main(int argc, char **agrv)
{
    //1
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    int gcd = getGcd(a, b);
    int lcm = getLcm(a, b);
    printf("GCD: %d\n", gcd);
    printf("LCM: %d\n", lcm);
    //4
    int c;
    printf("Enter a natural number: ");
    scanf("%d", &c);
    int nat = naturial(c);
    printf("Sum of digits natural number: %d\n", nat);

}

```

11laba main.c



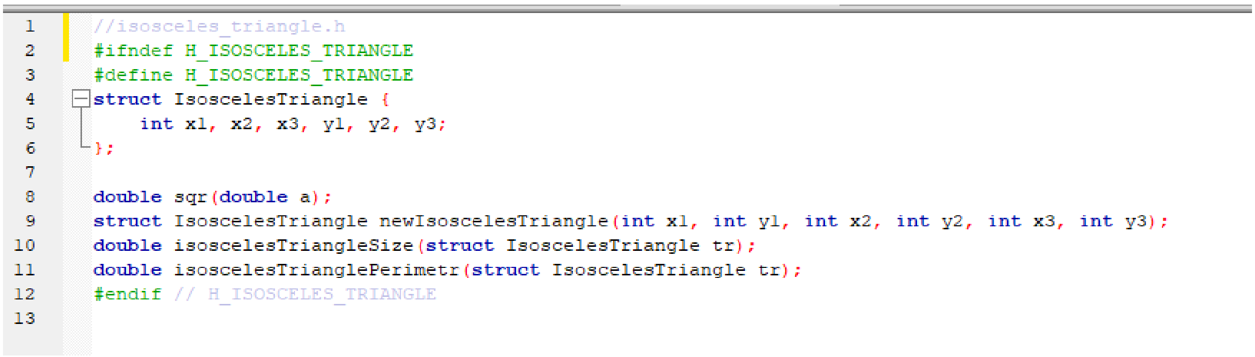
```c
//main.c
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include "C:\Users\Даурчик\Desktop\laba11\isosceles_triangle.h"

int main() {
    int x1, x2, x3, y1, y2, y3;
    scanf("%d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3);
    struct IsoscelesTriangle tr = newIsoscelesTriangle(x1, y1, x2, y2, x3, y3);
    printf("Perimetr is %lf\nSize is %lf", isoscelesTrianglePerimetr(tr), isoscelesTriangleSize(tr));
    return 0;
}
```

isosceles_triangle.c



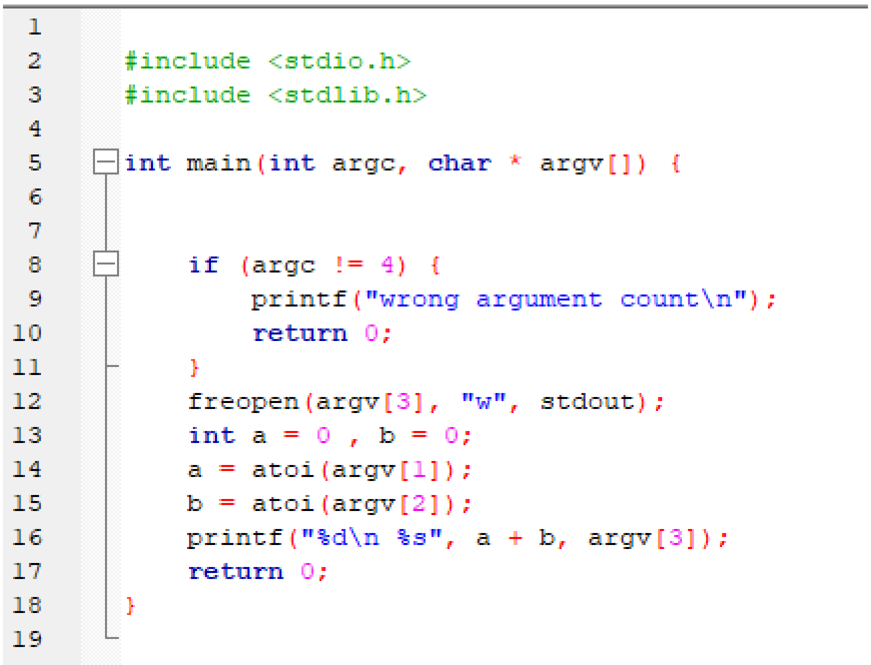
```c
//isosceles_triangle.c
#include <math.h>
#include "isosceles_triangle.h"
struct IsoscelesTriangle {
    int x1, x2, x3, y1, y2, y3;
};

struct IsoscelesTriangle newIsoscelesTriangle(int x1, int y1, int x2, int y2, int x3, in
    struct IsoscelesTriangle result;
    result.x1 = x1;
    result.y1 = y1;
    result.x2 = x2;
    result.y2 = y2;
    result.x3 = x3;
    result.y3 = y3;
    return result;
}

double sqr(double a) {
    return a * a;
}

double isoscelesTriangleSize(struct IsoscelesTriangle tr) {
    double a = sqrt(sqr(tr.x1 - tr.x2) + sqr(tr.y1 - tr.y2));
    double b = sqrt(sqr(tr.x3 - tr.x2) + sqr(tr.y3 - tr.y2));
    double c = sqrt(sqr(tr.x1 - tr.x3) + sqr(tr.y1 - tr.y3));
    double p = (a + b + c) / 2;
    return sqrt(p * (p - a) * (p - b) * (p - c));
}


double isoscelesTrianglePerimetr(struct IsoscelesTriangle tr) {
    double a = sqrt(sqr(tr.x1 - tr.x2) + sqr(tr.y1 - tr.y2));
    double b = sqrt(sqr(tr.x3 - tr.x2) + sqr(tr.y3 - tr.y2));
    double c = sqrt(sqr(tr.x1 - tr.x3) + sqr(tr.y1 - tr.y3));
    return a + b + c;
}

```

isosceles_triangle.h



```c
//isosceles_triangle.h
#ifndef H_ISOSCELES_TRIANGLE
#define H_ISOSCELES_TRIANGLE
struct IsoscelesTriangle {
    int x1, x2, x3, y1, y2, y3;
};

double sqr(double a);
struct IsoscelesTriangle newIsoscelesTriangle(int x1, int y1, int x2, int y2, int x3, int y3);
double isoscelesTriangleSize(struct IsoscelesTriangle tr);
double isoscelesTrianglePerimetr(struct IsoscelesTriangle tr);
#endif // H_ISOSCELES_TRIANGLE

```

12laba.



```c

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char * argv[]) {


    if (argc != 4) {
        printf("wrong argument count\n");
        return 0;
    }
    freopen(argv[3], "w", stdout);
    int a = 0 , b = 0;
    a = atoi(argv[1]);
    b = atoi(argv[2]);
    printf("%d\n %s", a + b, argv[3]);
    return 0;
}

```