ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ЛАБОРАТОРНОЙ РАБОТЕ №2

По дисциплине «Языки программирования»

ВАРИАНТ 6

Выполнил: ст. гр. ТКИ-141

Винтфельд Рина Дмитриевна

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2023

Оглавление

[1 Задача 2.1 3](#_Toc153409650)

[1.1 Формулировка задания 2.1 3](#_Toc153409651)

[1.2 Блок-схема алгоритма 2.1 4](#_Toc153409652)

[1.3 Текст программы на языке C 2.1 6](#_Toc153409653)

[1.4 Результаты выполнения программы 2.1 9](#_Toc153409654)

[1.5 Выполнение тестовых примеров 2.1 10](#_Toc153409655)

[1.6 Отметка о выполнении задания в веб-хостинге системы контроля версий 2.1 11](#_Toc153409656)

[2 Задача 2.2 12](#_Toc153409657)

[2.1 Формулировка задания 2.2 12](#_Toc153409658)

[2.2 блок-схема алгоритма 1.2 13](#_Toc153409659)

[2.3 Текст программы на языке C 1.2 16](#_Toc153409660)

[2.4 Результаты выполнения программы 1.2 18](#_Toc153409661)

[2.5 Выполнение тестовых примеров 1.2 19](#_Toc153409662)

[2.6 Отметка о выполнении задания в веб-хостинге системы контроля версий 1.2 20](#_Toc153409663)

[3 задача 1.3 21](#_Toc153409664)

[3.1 Формулировка задания 1.3 21](#_Toc153409665)

[3.2 Блок-схема алгоритма 1.3 22](#_Toc153409666)

[3.3 Текст программы на языке C 1.3 24](#_Toc153409667)

[3.4 Результаты выполнения программы 1.3 26](#_Toc153409668)

[3.5 Выполнение тестовых примеров 1.3 27](#_Toc153409669)

[3.6 Отметка о выполнении задания в веб-хостинге системы контроля версий 1.3 28](#_Toc153409670)

1. Задача 2.1
   1. Формулировка задания 2.1

Создать консольное приложение для решения задачи, представленной в Таблица 1, с использованием перечислимого типа. Выбор формулы вычисления зависит от пользователя. Данные для решения задачи так же вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Задача |
| 6 | Поменяйте между собой значения переменных *a* и *b*,   * воспользовавшись третьей переменной *с*; * без использования третьей переменной. |

* 1. Блок-схема алгоритма 2.1

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы используемых функций представлены ниже (Рисунок 2, Рисунок 3).

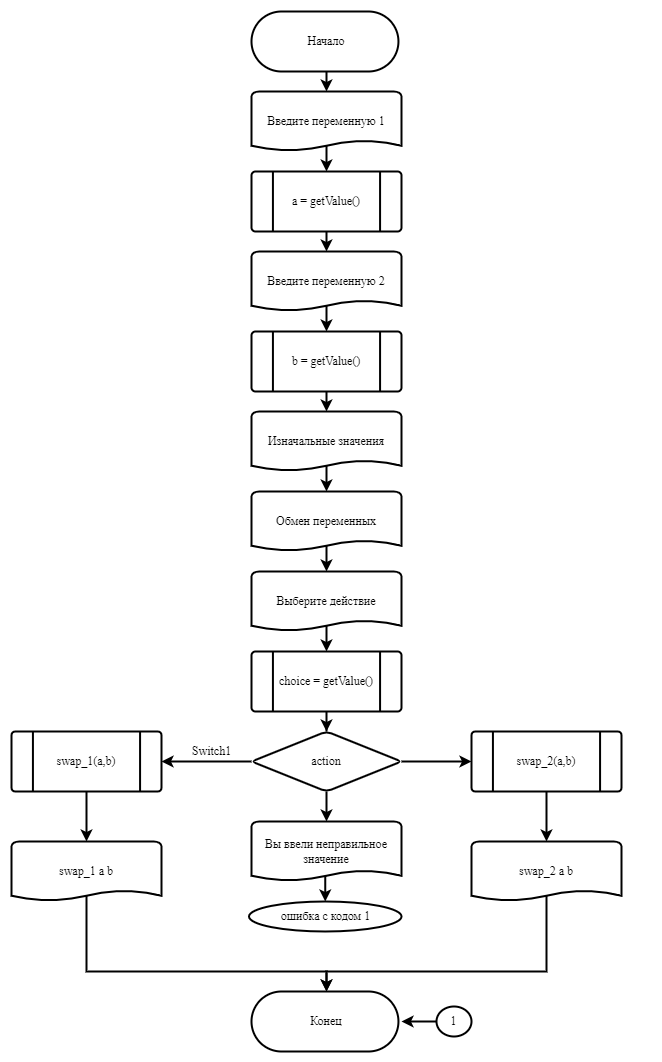


Рисунок 1 ­- Блок-схема основного алгоритма

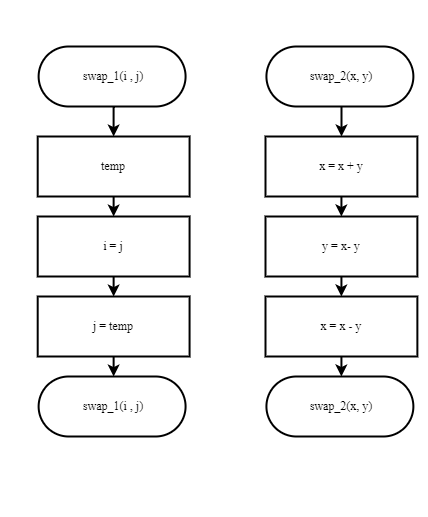


Рисунок 2 – Блок-схема используемых функций

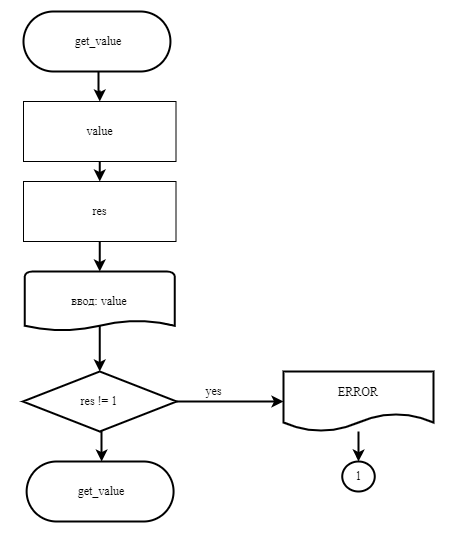


Рисунок 3 - Блок-схема используемой функций

* 1. Текст программы на языке C 2.1

#include <stdio.h>

#include <stdlib.h> //abort

#include <errno.h> //error

/\*

\* @brief Функция меняет между собой значения двух переменных при помощи третьей

\* @param i - первая переменная

\* @param j - вторая переменная

\*/

void swap\_1(double\* i, double\* j);

/\*

\* @brief Функция меняет между собой значения двух переменных

\* @param x - первая переменная

\* @param y - вторая переменная

\*/

void swap\_2(double\* x, double\* y);

/\*

\* @brief проверяет пользовательский ввод

\* @return введеное значение

\*/

double get\_value();

/\*

\* @brief Пользовательский ввод

\* @param Switch1 действие 1) обмен переменных с помощью 3 переменной

\* @param Switch2 действие 2) обмен переменных без помощи 3 переменной

\*/

enum Action{Switch1=1, Switch2};

/\*

\* @brief является точкой входа в программу

\* @return 0, в случае успеха

\*/

int main() {

printf\_s("Введите переменную 1\n");

double a = get\_value();

printf\_s("Введите переменную 2\n");

double b = get\_value();

printf\_s("Изначальные значения: a = %lf, b = %lf\n", a, b);

printf\_s("%d - Обмен переменных при помощи 3 переменной\n%d - Обмен переменных без помощи 3 переменной\n", Switch1, Switch2);

printf\_s("Выберите действие: ");

int choice = get\_value();

enum Action action = (enum Action)choice;

switch (action)

{

case Switch1:

swap\_1(&a, &b);

printf\_s("swap\_1 a = %.1lf, b = %.1lf\n", a, b);

break;

case Switch2:

swap\_2(&a, &b);

printf\_s("swap\_2 a = %.1lf, b = %.1lf\n", a, b);

break;

default:

printf\_s("Вы ввели неправильное значение");

return 1;

break;

}

return 0;

}

double get\_value()

{

double value;

int res = scanf\_s("%lf", &value);

if (res != 1)

{

errno = EIO;

perror("ERROR");

abort();

}

return value;

}

void swap\_1(double\* i, double\* j) {

double temp = \*i;

\*i = \*j;

\*j = temp;

}

void swap\_2(double \*x, double \*y){

\*x = \*x + \*y;

\*y = \*x - \*y;

\*x = \*x - \*y;

}

* 1. Результаты выполнения программы 2.1

Результаты выполнения программы представлены ниже (Рисунок 4, Рисунок 5).

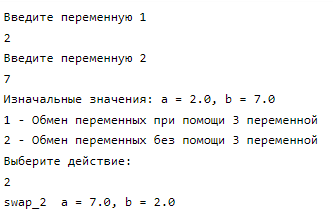


Рисунок 4 - Результат выполнения второго пункта программы

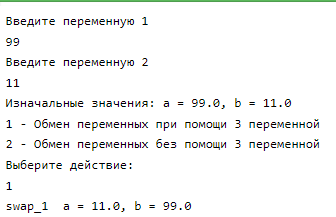


Рисунок 5 - Результат выполнения первого пункта программы

* 1. Выполнение тестовых примеров 2.1

В программе Paint выполнен тестовый пример. Результат его выполнения представлен ниже (Рисунок 6).

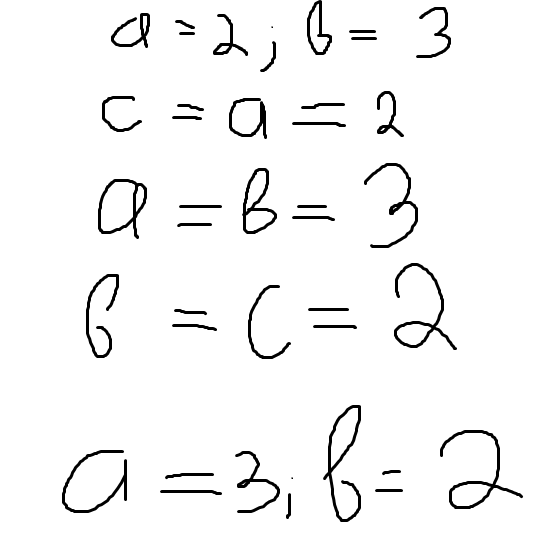


Рисунок 6 – Тестовый пример

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 2.1

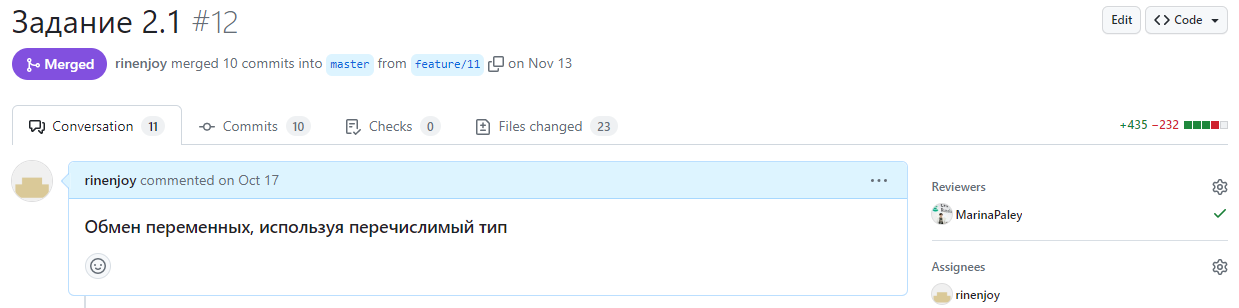


Рисунок 7 - Approved task 2.1

1. Задача 2.2
   1. Формулировка задания 2.2

Создать консольное приложение для решения задачи, представленной в таблице. Данные для решения вводит пользователь. Помните, что ввод необходимо проверять на правильность (только числа). Вывести результат вычислений на экран. При необходимости дополнить свой отчёт поясняющими формулами, помогающими решить задачу. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 2 - Исходные данные

|  |  |
| --- | --- |
| Вариант | Задание |
| 6 | Вычислить расстояние между двумя точками с заданными координатами. |

* 1. блок-схема алгоритма 1.2

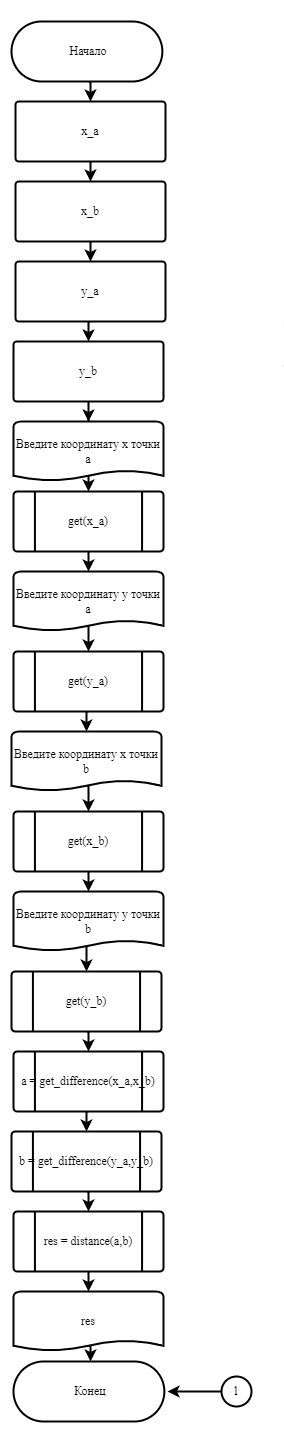
Блок-схема основного алгоритма представлена ниже (Рисунок 7). Блок-схемы используемых функций представлены ниже (Рисунок 8, Рисунок 9) 

Рисунок 7 - Блок-схема основного алгоритма

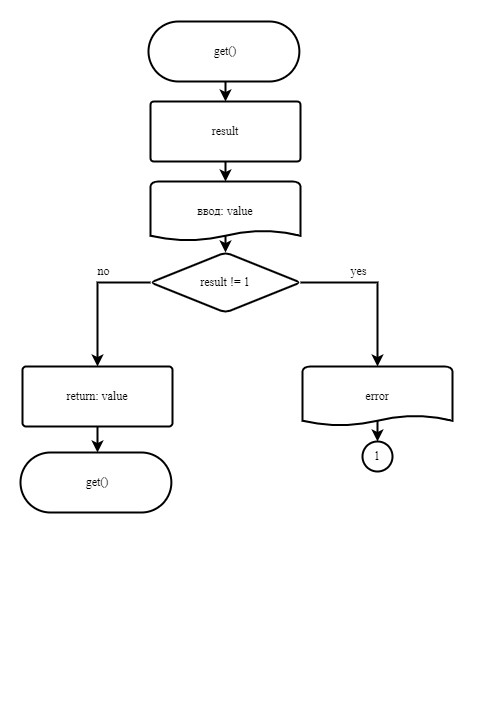


Рисунок 8 - Блок-схема используемой функции

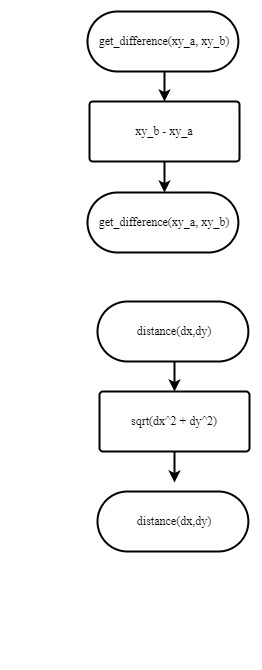


Рисунок 9 - Блок-схема используемой функции

* 1. Текст программы на языке C 1.2

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция получает одну из координат двух точек

\* @return возвращает разницу между координатами

\*/

double get\_difference(double xy\_a, double xy\_b);

/\*\*

\*@brief расчитывает расстояние между двумя заданными точками

\*@param функция ожидает два значения, полученные в результате работы функций get\_x и get\_y

\*@return возвращает длину расстояния между двух точек

\*/

double distance(double dx, double dy);

/\*\*

\*@brief проверяет на правильность пользовательский ввод

\*/

double get();

/\*\*

\*@brief является точкой входа в программу

\*@return возвращает 0, в случае успеха

\*/

int main()

{

puts("Введите координату x точки a: ");

double x\_a = get();

puts("Введите координату y точки a: ");

double y\_a = get();

puts("Введите координату x точки b: ");

double x\_b = get();

puts("Введите координату y точки b: ");

double y\_b = get();

double res = distance(get\_difference(x\_a, x\_b), get\_difference(y\_a, y\_b));

printf\_s("Растояние = %.2lf", res);

return 0;

}

double get()

{

double value;

int result = scanf\_s("%lf", &value);

if (result != 1)

{

puts("error");

abort();

}

return value;

}

double get\_difference(double xy\_a, double xy\_b)

{

return xy\_b - xy\_a;

}

double distance(double dx, double dy)

{

return sqrt(pow(dx, 2) + pow(dy, 2));

}

* 1. Результаты выполнения программы 1.2

Результаты выполнения программы представлены ниже (Рисунок 10, Рисунок 11, Рисунок 12).

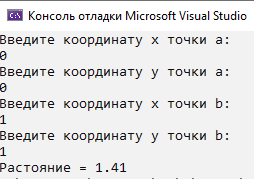


Рисунок 10 - Результаты выполнения программы

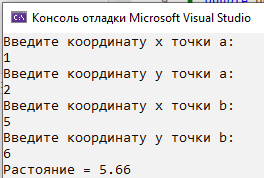


Рисунок 11 - Результат выполнения программы

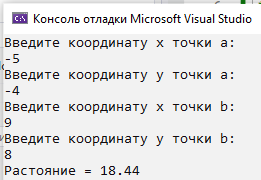


Рисунок 12 - Результат выполнения программы

* 1. Выполнение тестовых примеров 1.2

На сайте onlinemschool.com выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 13, Рисунок 14).

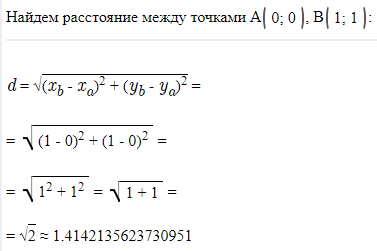


Рисунок 13 – Тестовый расчет

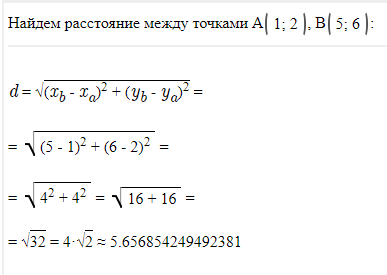


Рисунок 14 – Тестовый расчет

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 1.2

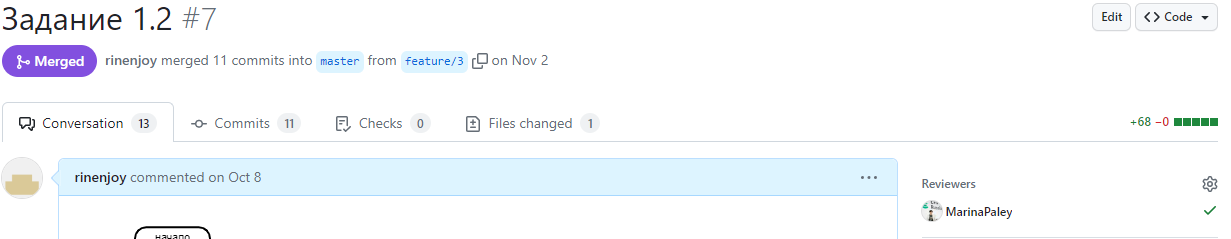


Рисунок 15 - Approved task 1.2

1. задача 1.3
   1. Формулировка задания 1.3

Создать консольное приложение для решения задачи, представленной в таблице. Данные для решения вводит пользователь. Помните, что ввод необходимо проверять на правильность (только числа). Вывести результат вычислений на экран. При необходимости дополнить свой отчёт поясняющими формулами, помогающими решить задачу. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 3 - Исходные данные

|  |  |
| --- | --- |
| Вариант | Задание |
| 6 | Рассчитайте давление, производимое бетонной плитой, масса которой равна *m* кг, а площадь опоры *S* м2. |

* 1. Блок-схема алгоритма 1.3

Блок-схема основного алгоритма представлена ниже (Рисунок 16). Блок-схемы используемых функций представлены ниже (Рисунок 17, Рисунок 18).

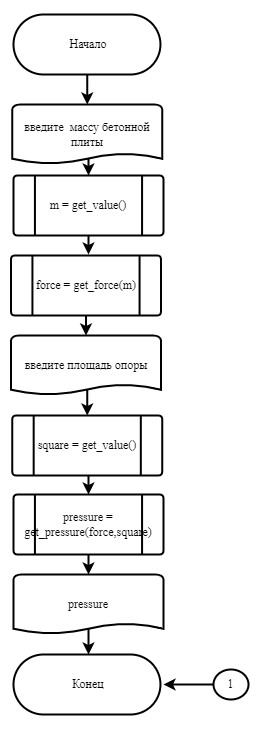


Рисунок 16 - Блок-схема основного алгоритма

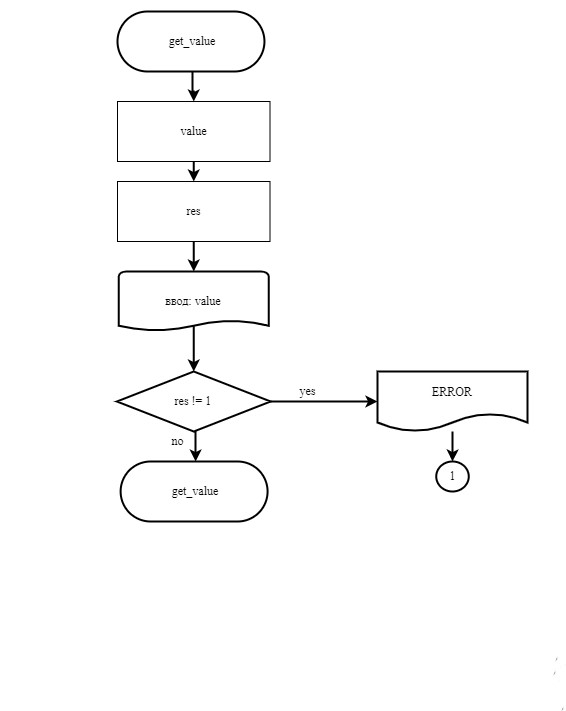


Рисунок 17 - Блок-схема используемой функции

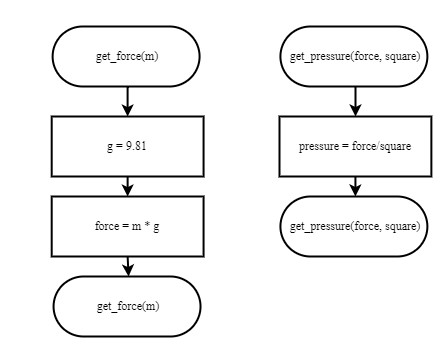


Рисунок 18 - Блок-схемы используемых функций

* 1. Текст программы на языке C 1.3

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\*

\* @brief Функция получает пользовательское значение

\*/

double get\_value();

/\*

\* @brief Функция вычисляет силу force

\* @param m - масса бетонной плиты [кг]

\* @return сила force [Н]

\*/

double get\_force(double m);

/\*

\* @brief Функция вычисляет давление, производимое бетонной плитой

\* @param force - сила давления бетонной плиты

\* @param square - площадь опоры

\* @return давление pressure [Па]

\*/

double get\_pressure(double force, double square);

/\*

\* @brief является точкой входа в программу

\* @return 0, в случае успеха

\*/

int main() {

puts("Введите массу (m) бетонной плиты [кг]: ");

double m = get\_value();

puts("Введите площадь опоры [кв.м] : ");

printf\_s("Давление, производимое бетонной плитой = %.2lf [Па]", get\_pressure(get\_force(m), get\_value()));

return 0;

}

double get\_value()

{

double value;

double res = scanf\_s("%lf", &value);

if (res != 1)

{

puts("Error");

abort();

}

return value;

}

double get\_force(double m)

{

const double g = 9.81;

return (m \* g);

}

double get\_pressure(double force, double square)

{

return (force / square);

}

* 1. Результаты выполнения программы 1.3

Результаты выполнения программы представлены ниже (Рисунок 19, Рисунок 20).

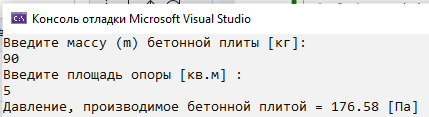


Рисунок 19 - Результаты выполнения программы

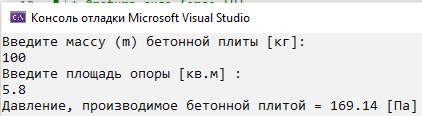


Рисунок 20 - Результаты выполнения программы

* 1. Выполнение тестовых примеров 1.3

На сайте planetcalc.ru выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 21, Рисунок 22).

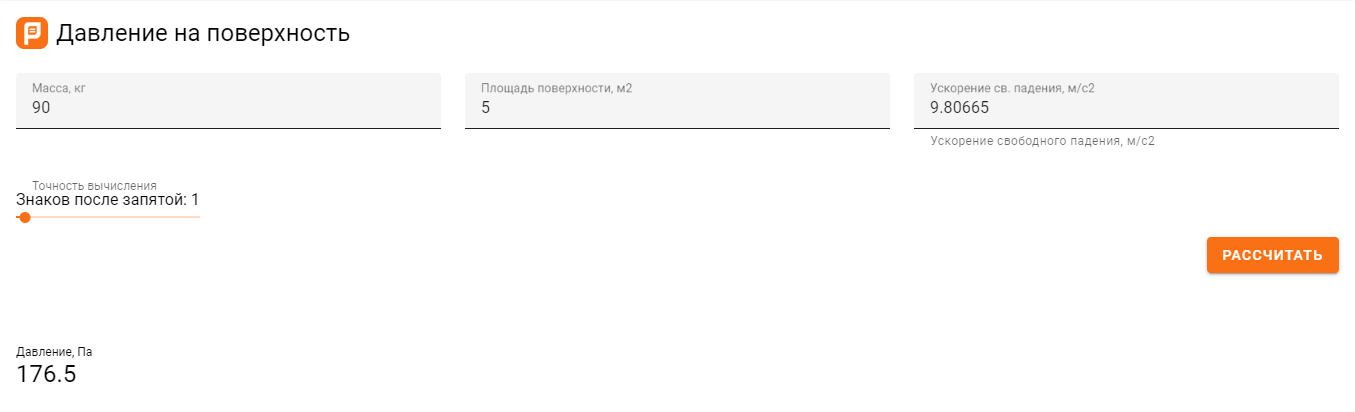


Рисунок 21 – Тестовый расчет

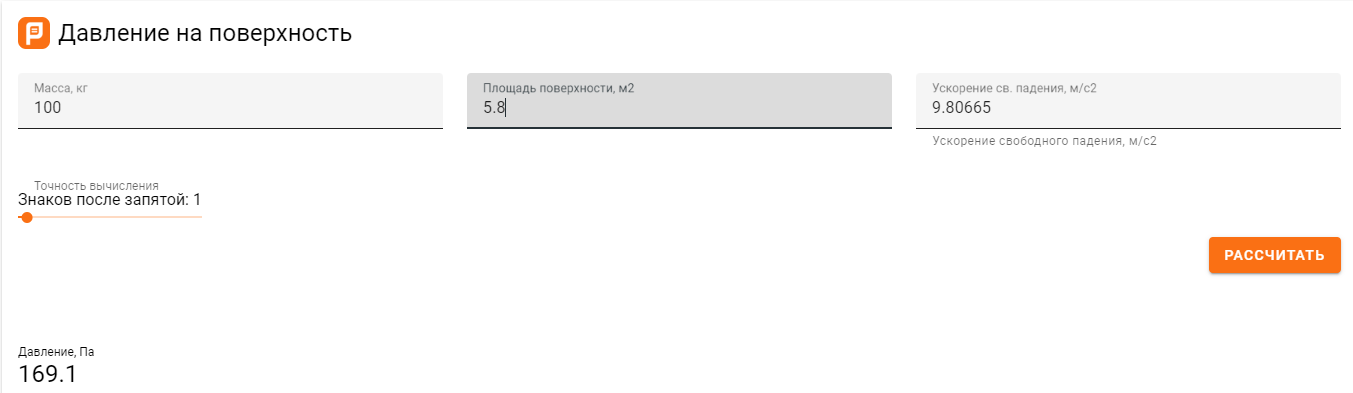


Рисунок 22 – Тестовый расчет

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий 1.3

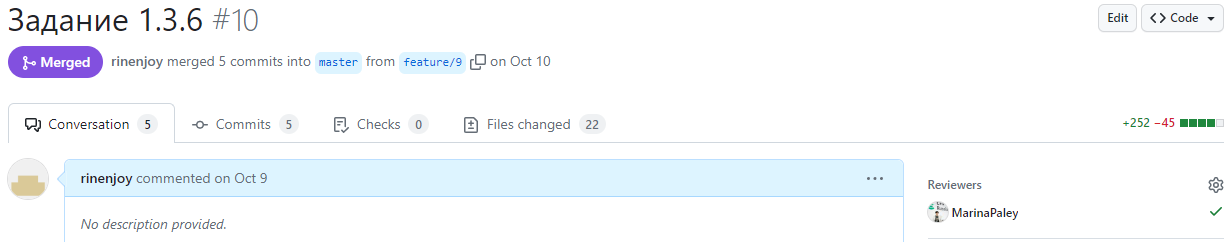


Рисунок 23 - Approved task 1.3