

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ(ТУСУР)

Кафедра комплексной информационной безопасности электронно-
вычислительных систем (КИБЭВС)

Функции и рекурсия

Отчет по лабораторной работе №4
по дисциплине “Основы программирования”

Выполнил
Студент гр. 728-2
_____ Геворгян Д. Р.
28 марта 2019

Доцент кафедры БИС
_____ Харченко С. С.
“__” _____ 2019 г.

1. Введение

Цель работы: создание блок-схем алгоритма и программ с использованием функций, созданных пользователем.

Вариант 7

Задача 1: Рассчитать полную поверхность треугольной пирамиды, заданной координатами своих вершин. Для определения расстояния между точками в пространстве оформить функцию.

Задача 2: Рассчитать биномиальные коэффициенты по формулам $C_n^0 = 1$ для $n \geq 0$,

$$C_n^m = 0 \text{ для } n > m \geq 0,$$

$$C_n^m = C_{n-1}^{m-1} + C_{n-1}^m \text{ для } n \geq m > 0$$

2. Теоретическая часть

В C++ можно создавать пользовательские функции. Собственно говоря, мы с самого начала создавали функции: `main()` – не что иное, как главная функция пользователя.

В библиотеке C++ имеется немало встроенных функций. Они размещены в отдельных заголовочных файлах (тех самых, которые подключаются с помощью `#include`). Заголовочный файл имеет имя и расширение `.h`.

Мы уже использовали заголовочные файлы: `iostream.h` – содержит функции для ввода-вывода; `math.h` – содержит математические функции.

Для создания функции пользователя (и ее последующего исполнения) необходимо ее описать:

- Задать прототип функции
тип имя_функции (параметры);
это делается там же, где описываются переменные, т.е. до заголовка `main()` (обратим внимание, что в конце записи прототипа точка с запятой ставится обязательно!);
- Собственно описать функцию
тип имя_функции (параметры)
{ тело функции }

Такое описание делается вне тела другой функции, в том числе и функции `main()` (как правило, за закрывающей фигурной скобкой главной функции).

В конце описания (после фигурной скобки) точка с запятой не ставится. Функция может иметь только одно возвращаемое значение, которое определяется служебным словом `return` (возвращаемое_значение), которое размещается в конце функции (перед закрывающей фигурной скобкой).

Внутри функции могут быть описаны свои локальные переменные.

В C++ допустима рекурсия – обращение к самой себе.

3. Ход работы

3.1. Решение задачи 1

Для начала составим алгоритм:

Функция для нахождения расстояния между точками:

A1. Введём функцию distance для набора переменных (X1, Y1, Z1, X2, Y2, Z2);

A2. $d \leftarrow \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2 + (Z1 - Z2)^2}$

A3. Возвращаем значение переменной d.

Функция для нахождения площади треугольника:

B1. Введём функцию square для набора переменных (a, b, c);

B2. $p \leftarrow \frac{a+b+c}{2}$;

B3. $s \leftarrow \sqrt{p * (p - a) * (p - b) * (p - c)}$

B4. Возвращаем значение переменной s.

Главная функция:

C1. Вывод “Введите координаты точки A”;

C2. Ввод Xa, Ya, Za;

C3. Вывод “Введите координаты точки B”;

C4. Ввод Xb, Yb, Zb;

C5. Вывод “Введите координаты точки C”;

C6. Ввод Xc, Yc, Zc;

C7. Вывод “Введите координаты точки D”;

C8. Ввод Xd, Yd, Zd;

C9. $d_{AB} \leftarrow \text{distance}(Xa, Ya, Za, Xb, Yb, Zb)$;

C10. $d_{AC} \leftarrow \text{distance}(Xa, Ya, Za, Xc, Yc, Zc)$;

- C11. $d_{BC} \leftarrow \text{distance}(X_b, Y_b, Z_b, X_c, Y_c, Z_c);$
 C12. $d_{AD} \leftarrow \text{distance}(X_a, Y_a, Z_a, X_d, Y_d, Z_d);$
 C13. $d_{CD} \leftarrow \text{distance}(X_c, Y_c, Z_c, X_d, Y_d, Z_d);$
 C14. $d_{BD} \leftarrow \text{distance}(X_b, Y_b, Z_b, X_d, Y_d, Z_d);$
 C15. $s_{ABC} \leftarrow \text{square}(d_{AC}, d_{AB}, d_{BC});$
 C16. $s_{ACD} \leftarrow \text{square}(d_{AC}, d_{AD}, d_{CD});$
 C17. $s_{ABD} \leftarrow \text{square}(d_{AB}, d_{AD}, d_{BD});$
 C18. $s_{BCD} \leftarrow \text{square}(d_{BC}, d_{BD}, d_{CD});$
 C19. $S \leftarrow s_{ABC} + s_{ACD} + s_{ABD} + s_{BCD};$
 C20. Вывод “Полная поверхность треугольной пирамиды равна ”, S .

Теперь представим решение задачи в виде блок-схемы:

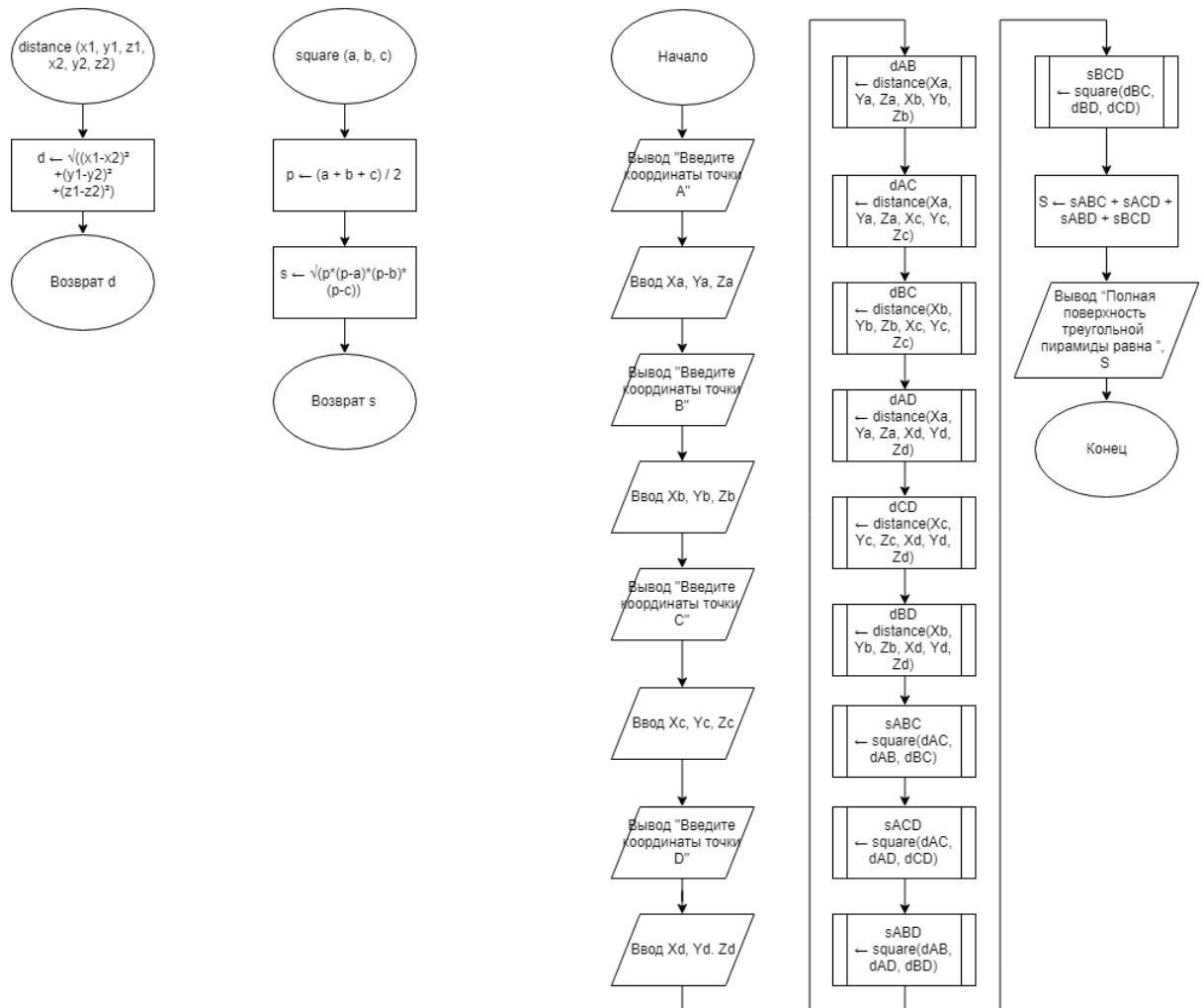


Рисунок 1 – Блок-схема задачи 1

Далее напишем код программы на языке C++:

```
#include "pch.h"
#include "math.h"
#include <iostream>
using namespace std;

double distance(double X1, double Y1, double Z1, double
X2, double Y2, double Z2)
{
    double d;
    d = sqrt((pow((X1 - X2), 2)) + (pow((Y1 - Y2), 2)) +
(pow((Z1 - Z2), 2)));
    return (d);
}

double square(double a, double b, double c)
{
    double p, s;
    p = (a + b + c) / 2;
    s = sqrt(p*(p - a)*(p - b)*(p - c));
    return (s);
}

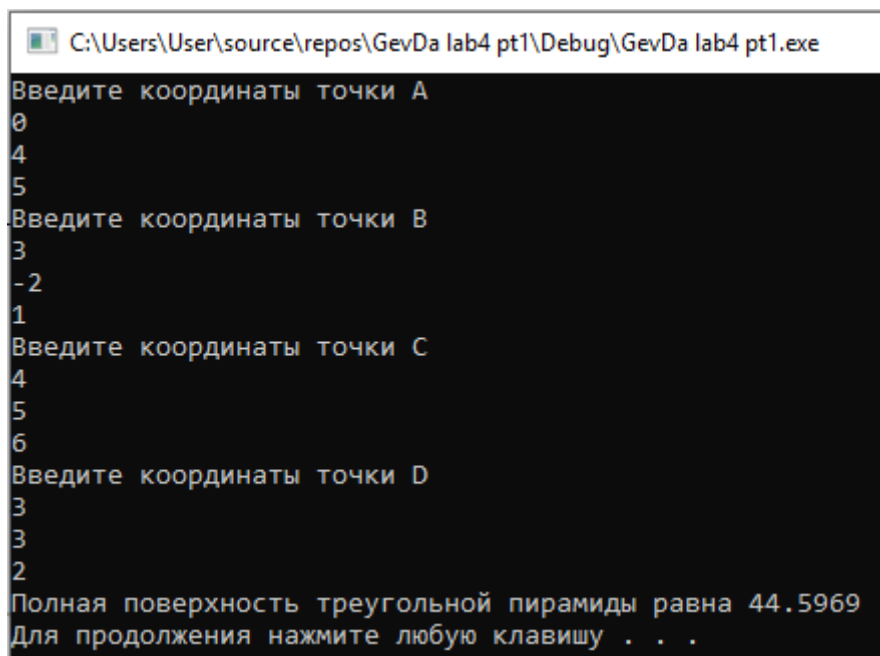
int main()
{
    setlocale(LC_ALL, "Russian");
    double Xa, Ya, Za, Xb, Yb, Zb, Xc, Yc, Zc, Xd, Yd, Zd;
    cout << "Введите координаты точки A \n";
    cin >> Xa >> Ya >> Za;
    cout << "Введите координаты точки B \n";
    cin >> Xb >> Yb >> Zb;
    cout << "Введите координаты точки C \n";
    cin >> Xc >> Yc >> Zc;
    cout << "Введите координаты точки D \n";
    cin >> Xd >> Yd >> Zd;
    double dAB, dAC, dBC, dAD, dCD, dBD;
    dAB = distance(Xa, Ya, Za, Xb, Yb, Zb);
    dAC = distance(Xa, Ya, Za, Xc, Yc, Zc);
    dBC = distance(Xb, Yb, Zb, Xc, Yc, Zc);
    dAD = distance(Xa, Ya, Za, Xd, Yd, Zd);
    dCD = distance(Xc, Yc, Zc, Xd, Yd, Zd);
```



```

dBD = distance(Xb, Yb, Zb, Xd, Yd, Zd);
double sABC, sACD, sABD, sBCD;
sABC = square(dAC, dAB, dBC);
sACD = square(dAC, dAD, dCD);
sABD = square(dAB, dAD, dBD);
sBCD = square(dBC, dBD, dCD);
double S;
S = sABC + sACD + sABD + sBCD;
cout << "Полная поверхность треугольной пирамиды равна
" << S;
system("pause");
return 0;
}

```



```

C:\Users\User\source\repos\GevDa lab4 pt1\Debug\GevDa lab4 pt1.exe
Введите координаты точки A
0
4
5
Введите координаты точки B
3
-2
1
Введите координаты точки C
4
5
6
Введите координаты точки D
3
3
2
Полная поверхность треугольной пирамиды равна 44.5969
Для продолжения нажмите любую клавишу . . .

```

Рисунок 2 – Результат работы программы 1

3.2. Решение задачи 2

Для начала составим алгоритм:

Функция формулы:

- A1. Введём функцию bin для набора переменных (m, n) ;
- A2. Если $(n \geq 0)$ и $(m = 0)$, то выполняем A3, иначе A4;
- A3. Возвращаем (1);
- A4. Если $(m > n)$ и $(n \geq 0)$, то выполняем A5, иначе A6;
- A5. Возвращаем (0);
- A6. Возвращаем $(\text{bin}(m-1, n-1) + \text{bin}(m, n-1))$.

Главная функция:

- B1. Ввод x, y ;
- B2. Вывод $\text{bin}(x, y)$;
- B3. Возвращаем (0).

Теперь представим решение задачи в виде блок-схемы:

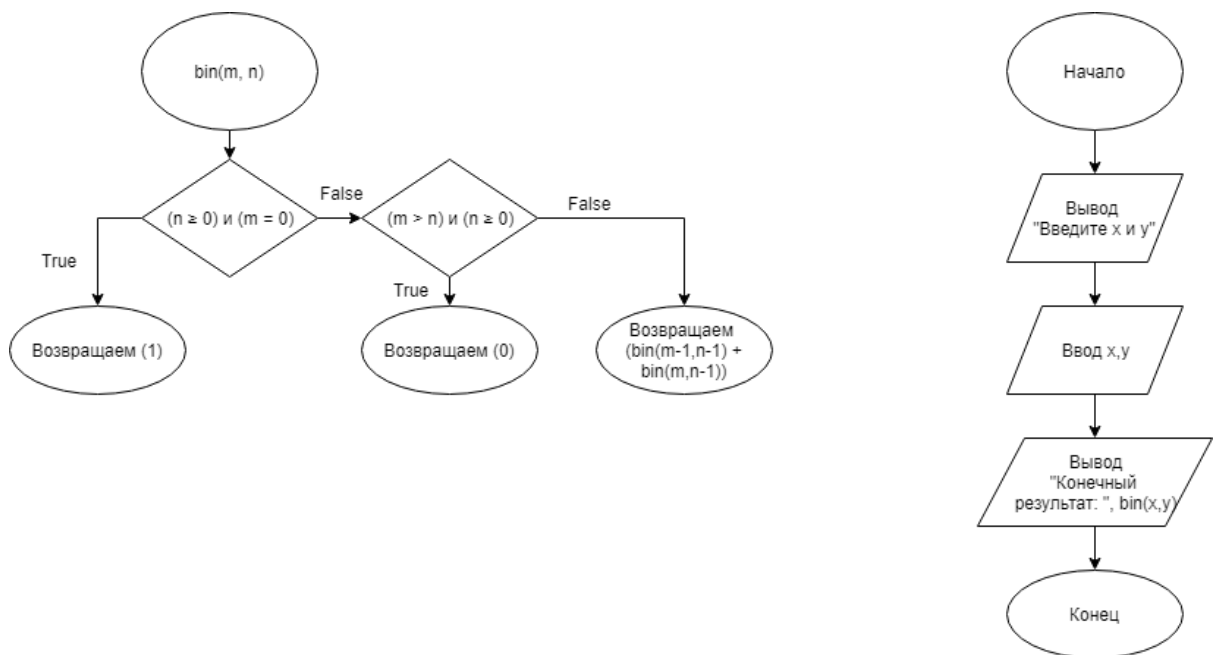


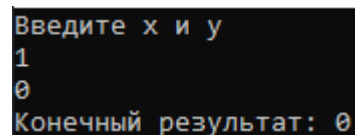
Рисунок 3 – Блок-схема задачи 2

Далее представим программу на языке C++:

```
#include "pch.h"
#include <stdio.h>
#include <iostream>
using namespace std;

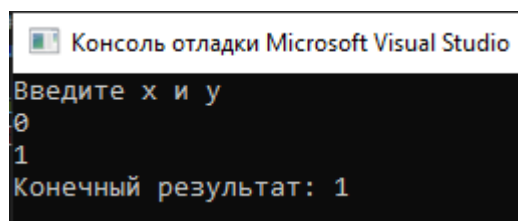
int bin(int m, int n)
{
    if ((n >= 0) && (m == 0))
        return 1;
    if ((m > n) && (n >= 0))
        return 0;
    else
        return (bin(m - 1, n - 1) + bin(m, n - 1));
}

int main()
{
    int x, y;
    cin >> x >> y;
    cout << bin(x, y) << "\n";
    return 0;
}
```



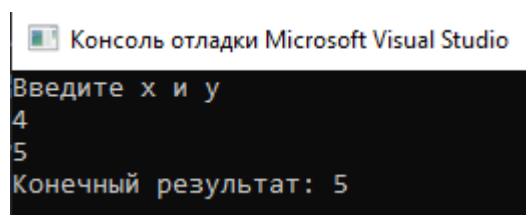
```
Введите x и y
1
0
Конечный результат: 0
```

Рисунок 4 – Результат работы программы при x=1 и y=0



```
Консоль отладки Microsoft Visual Studio
Введите x и y
0
1
Конечный результат: 1
```

Рисунок 5 – Результат работы программы при x=0 и y=1



```
Консоль отладки Microsoft Visual Studio
Введите x и y
4
5
Конечный результат: 5
```

Рисунок 6 – Результат работы программы при x=4 и y=5

4. Заключение

В процессе выполнения лабораторной работы были получены навыки создания блок-схем алгоритма и программ с использованием функций, созданных пользователем.

Отчёт был написан согласно ОС ТУСУР 2013.