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 ,

* 1. Теоретическая часть

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

В библиотеке С++ 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 слово return (возвращаемое\_значение), которое размещается в конце функции (перед закрывающей фигурной скобкой).

Внутри функции могут быть описаны свои локальные переменные.

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

* 1. Ход работы

3.1. Решение задачи 1

Для начала составим алгоритм:

Функция для нахождения расстояния между точками:

А1. Введём функцию distance для набора переменных (X1, Y1, Z1, X2, Y2, Z2);  
A2. d 🡸   
А3. Возвращаем значение переменной d.

Функция для нахождения площади треугольника:

В1. Введём функцию square для набора переменных (a, b, c);  
B2. p 🡸 ;  
B3. s 🡸  
В4. Возвращаем значение переменной s.

Главная функция:

C1. Вывод “Введите координаты точки А”;  
C2. Ввод Xa, Ya, Za;  
C3. Вывод “Введите координаты точки В”;  
С4. Ввод Xb, Yb, Zb;  
C5. Вывод “Введите координаты точки С”;  
С6. Ввод Xc, Yc, Zc;  
C7. Вывод “Введите координаты точки D”;  
С8. Ввод Xd, Yd, Zd;  
C9. dAB 🡸 distance(Xa, Ya, Za, Xb, Yb, Zb);  
C10. dAC 🡸 distance(Xa, Ya, Za, Xc, Yc, Zc);  
C11. dBC 🡸 distance(Xb, Yb, Zb, Xc, Yc, Zc);  
C12. dAD 🡸 distance(Xa, Ya, Za, Xd, Yd, Zd);  
C13. dCD 🡸 distance(Xc, Yc, Zc, Xd, Yd, Zd);  
C14. dBD 🡸 distance(Xb, Yb, Zb, Xd, Yd, Zd);  
C15. sABC 🡸 square(dAC, dAB, dBC);  
C16. sACD 🡸 square(dAC, dAD, dCD);  
C17. sABD 🡸 square(dAB, dAD, dBD);  
C18. sBCD 🡸 square(dBC, dBD, dCD);  
C19. S 🡸 sABC + sACD + sABD + sBCD;  
С20. Вывод “Полная поверхность треугольной пирамиды равна ”, S.

Теперь представим решение задачи в виде блок-схемы:

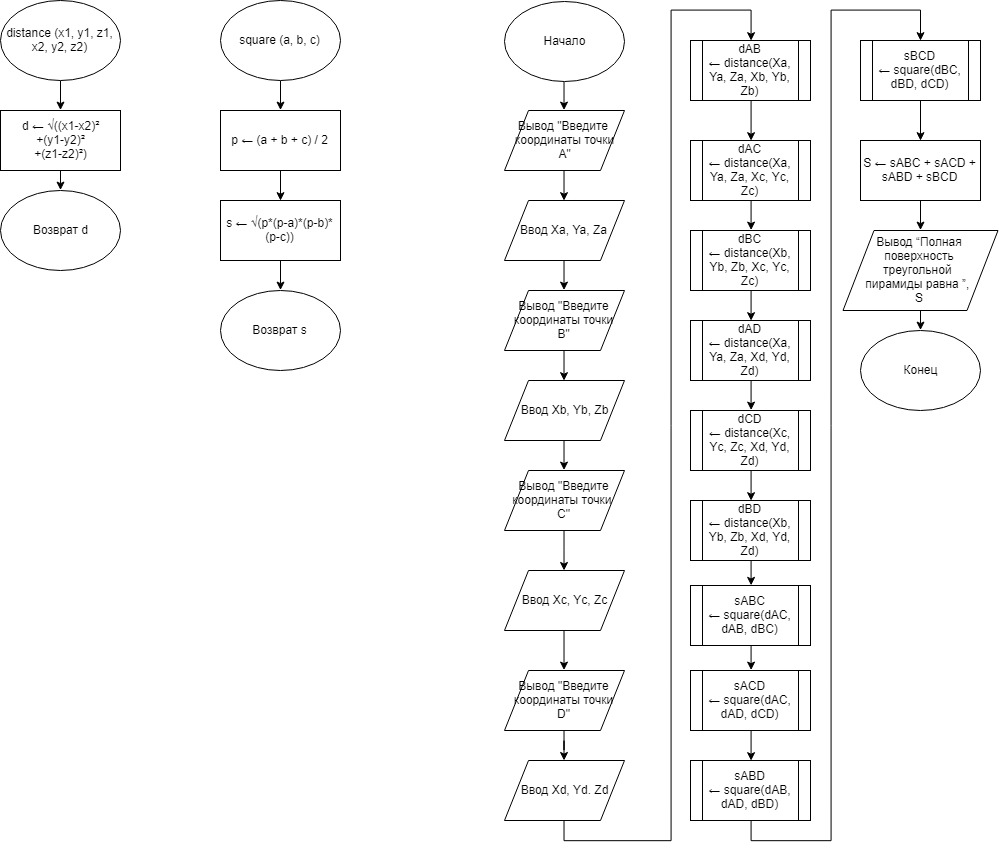


Рисунок 1 – Блок-схема задачи 1

Далее напишем код программы на языке С++:

#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 А \n";

cin >> Xa >> Ya >> Za;

cout << "Введите координаты точки В \n";

cin >> Xb >> Yb >> Zb;

cout << "Введите координаты точки С \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;

}

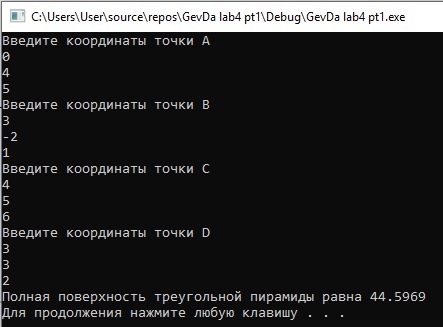


Рисунок 2 – Результат работы программы 1

3.2. Решение задачи 2

Для начала составим алгоритм:

Функция формулы:

А1. Введём функцию bin для набора переменных (m,n);  
А2. Если (n ≥ 0) и (m = 0), то выполняем А3, иначе А4;  
А3. Возвращаем (1);  
А4. Если (m > n) и (n ≥ 0), то выполняем А5, иначе А6;  
А5. Возвращаем (0);  
А6. Возвращаем (bin(m-1, n-1)+bin(m,n-1)).

Главная функция:

B1. Ввод x, y;  
B2. Вывод bin(x,y);  
B3. Возвращаем (0).

Теперь представим решение задачи в виде блок-схемы:

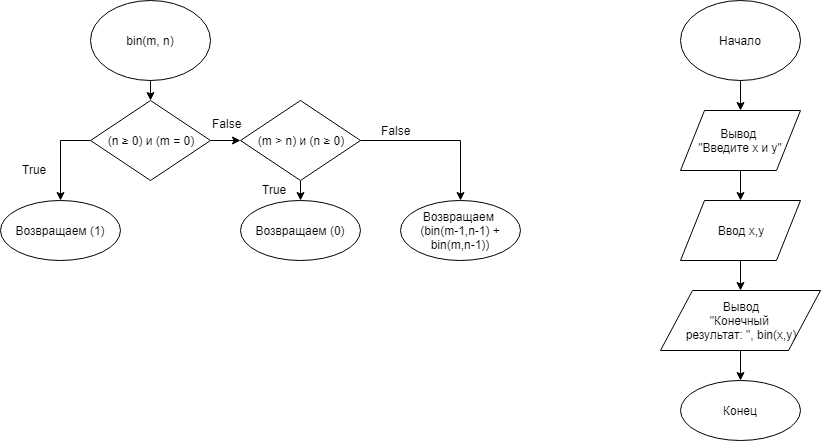


Рисунок 3 – Блок-схема задачи 2

Далее представим программу на языке С++:

#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;

}

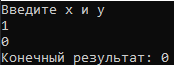


Рисунок 4 – Результат работы программы при x=1 и у=0

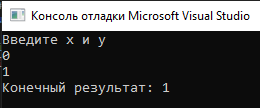


Рисунок 5 – Результат работы программы при x=0 и y=1

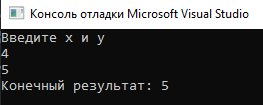


Рисунок 6 – Результат работы программы при x=4 и y=5

* 1. 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.