МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСІТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТІТУТ»

КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

**Лабораторна робота № 6**

з дисципліни «Алгоритмізація та програмування»

на тему: "Алгоритми послідовності (лінійної) структури»

Варіант 16

**Виконала:**

Студент групи БС-32

Хоміцевич Микита

**Перевірила:**

викл.

Матвійчук А.О.

Київ-2014

**Задание**

****

**□ Лабораторна робота виконана без зауважень**

**□ Лабораторна робота має зауваження:**

**□ присутні зауваження до блок-схем:**

**□ виконані не за стандартом**

**□ не відповідають коду**

**□ присутній код**

**□ інші зауваження:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**□присутні зауваження до коду:**

**□задача завдання вирішена хибно**

**□ код програми не компілюється**

**□ використано глобальні змінні**

**□ недостатня декомпозиція на функціонуванні користувача**

**□ статичні змінні при роботі з масивами**

**□ оформлення коду**

**□присутні зайві символи «{»та «}»**

**□інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**□невірні відповіді на запитання:**

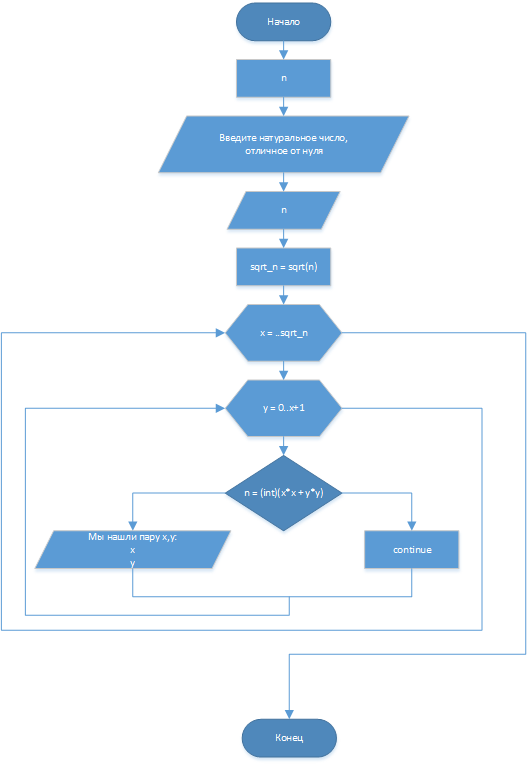
**□№1 □№2 □№3**

**□маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок – схема**

****

**Программный код**

#include <iostream>

#include <cmath>

#include <locale.h>

#include <conio.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

int n;

cout << "Введите натуральное число, отличное от нуля: ";

cin >> n;

double sqrt\_n = sqrt(n);

for (int x = 0; x < sqrt\_n; ++x)

for (int y = 0; y < x+1; ++y) {

if (n == (int)(x\*x + y\*y)) {

cout << "Мы нашли пару 'x,y': " << x << " " << y << endl;

}

else {

continue;

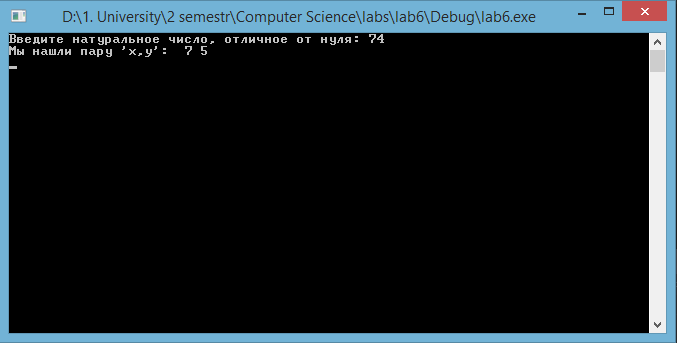
}

}

\_getch();

}

**Скриншот работы программы**

****

**Ответы на контрольные вопросы**

1. **Для каких целей используют циклы в программировании?**

**Цикл** — разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций. Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).

В основном циклы в программировании используются при каких-то взаимодествиях или операциях с множествами (либого рода). То есть, мы можем использовать цикл для того, что бы посчитать факториал числа, заданого самостоятельно или мы можем изменить все буквы в строке с маленьких на большие. В зависимости от наших изначальных условий, мы можем использовать три основных вида цикла (а так же еще 4 вид – совместный цикл, который, конкретно в языке С++, появился в новом стандарте С++11): цикл со счетчиком(как правило, удобен тем, если нам необходимо что-то посчитать или изменить с определенным упорядоченым набором данных), цикл с предусловием (когда мы задаем какое-то условие и ожидаем какого-то конечно ответа, при этом точно понимаю, является ли наше услвоие истиным или ложным) и цикл с постусловием (не важно, верно заданное изначально условие или нет – цикл выполнит все вложеные действия хотя бы 1 раз).

1. **Что такое цикл с параметром, формат его использования?**

Цикл с параметром (со счётчиком) — цикл, в котором некоторая переменная изменяет своё значение от заданного начального значения до конечного значения с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз. В большинстве процедурных языков программирования реализуется оператором **for**, в котором указывается счётчик (так называемая «переменная цикла»), требуемое количество проходов (или граничное значение счётчика) и, возможно, шаг, с которым изменяется счётчик. Например, в языке С и С++ данный цикл оформляется так:

for (statement 1; statement 2; statement 3)

{

//body of loop

//…

}

1. **Что такое цикл с предусловием, формат его использования?**

Цикл с предусловием — цикл, который выполняется пока истинно некоторое условие, указанное перед его началом. Это условие проверяется **до** выполнения тела цикла, поэтому тело может быть не выполнено ни разу (если условие с самого начала ложно). В большинстве процедурных языков программирования реализуется оператором **while**, отсюда его второе название — while-цикл.

На языке С и С++ имеет такую конструкцию:

while (<condition>)

{

// body of loop

}

1. **Что такое цикл с постусловием, формат его использования?**

**Цикл с постусловием**— цикл, в котором условие проверяется **после** выполнения тела цикла. Отсюда следует, что тело **всегда выполняется** хотя бы один раз.

В языке С++ этот цикл выглядит следующим образом:

do

{

// body of loop

}

while (<condition>)

1. **В чем является схожесть и разница между циклами с пред- и постусловиями?**

Схожесть:

* Задается определенное условие (условия), при которых цикл будет выполнятся.
* Оба цикла будут полностью выполнены, только если условие, которое будет приниматься на входе в цикл, будет = true.

Разница:

* Если условие, заданное в цикле, не является истинным, то цикл с предусловием не будет вполнятся совсем, а цикл с постусловием выполнится 1 раз.

1. **Что такое вложенные циклы?**

Существует возможность организовать цикл внутри тела другого цикла. Такой цикл будет называться *вложенным циклом*. Вложенный цикл по отношению к циклу в тело которого он вложен будет именоваться *внутренним циклом*, и наоборот цикл в теле которого существует вложенный цикл будет именоваться *внешним* по отношению к вложенному. Внутри вложенного цикла в свою очередь может быть вложен еще один цикл, образуя следующий *уровень вложенности* и так далее. Количество уровней вложенности, как правило, не ограничивается.

Полное число исполнений тела внутреннего цикла не превышает произведения числа итераций внутреннего и всех внешних циклов. Например взяв три вложенных друг в друга цикла, каждый по 10 итераций, получим 10 исполнений тела для внешнего цикла, 100 для цикла второго уровня и 1000 в самом внутреннем цикле.

1. **Для чего нужен оператор безадресного перехода break?**

Оператор break очень полезен при выплнении циклов, когда нам при определнном условии необходимо досрочно выйти из цикла.

1. **Для чего нужен оператор безадресного перехода continue?**

Оператор continue очень полезен, если нам надо использовать в какой-либо ветки условия какое-то действие (т.к. нельзя оставлять тело условия пустым), но при этом, что бы оно никак не влияло на данные самой программы.