Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий комплекс  
«Інститут прикладного системного аналізу»

Відділення другої вищої та післядипломної освіти

**ЛАБОРАТОРНА РОБОТА №1**

(варіант 10)

з курсу «Програмування»

на тему «Використання основних керуючих конструкцій»

Виконав: студент 3-го курсу

групи ІС-зп71

Бутузов О.В.

Прийняв: викл. Древаль М.М.

Захищено з оцінкою

« » 2017 р.

Київ – 2017

# Умова завдання

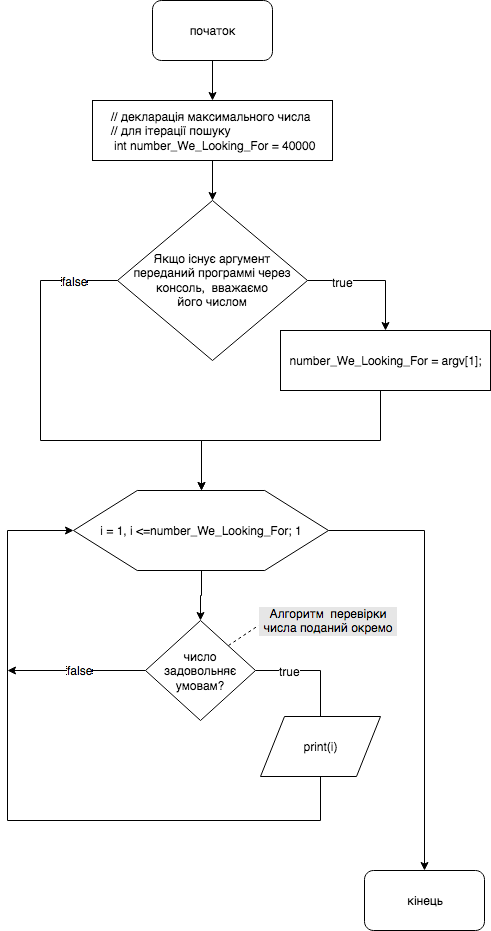
Дано натуральне число n. Серед чисел *1, ..., n* знайти такі числа, запис яких співпадає з останніми цифрами запису їх квадрату.

Наприклад:

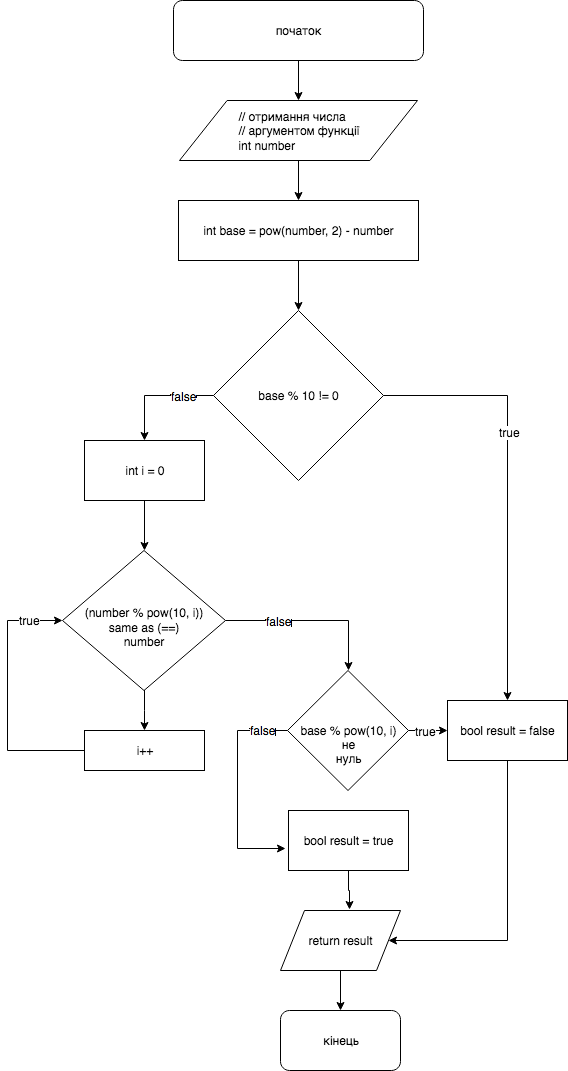
* 6 (62 = 36)
* 25 (252 = 625

# Блок Схема алгоритмів

Блок-схема алгоритму роботи програми:



Алгоритм валідації числа умовам нашої задачі



# Розвязання задачі

Я підходив до розвязання нашої задачі з математичної точки зору і робота над об’єктом – числом і його квадратом велась в десятичній системі числення. Як видно з више-приведеної блок схеми робота над числом проводилась в кілька етапів:

* По-перше число підносилось до квадрату і ввже від нього віднімалось число ( по суті проходила операція множення n на n-1), якщо остача від ділення цього числа на 10 була не нульовою – ми могли з впевненістю стверджувати що наше число аж ніяк не задовольняє умовам завдання.
* Наступним кроком було знайти – розряд числа яке ми підносили в квадрат, - цього ми досягли використовуючи цикл з передумовою (яка перевіряла чи буде остача від ділення нашого числа n на 10 в степені i дорівнювати нашому числу, як тільки ця умова виконається ми знайшли розряд нашого числа, “i” само собою тілі циклу)
* Останнім кроком в нашому розвязку буде фінальна перевірка ділення нашого базового числа ((n^2)-n або n\*(n -1)) на 10 в степені нашого розряду для знаходження остачі, наявність якої вказуватиме що число не відпомідає умовавм задачі.

# Лістинг програми

#include <iostream> /\* std::cout \*/

#include <stdlib.h> /\* atoi \*/

#include <math.h> /\* pow \*/

bool NaturalRootedNumber(int number);

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

// default number, in order to simplify development.

int number\_We\_Looking\_For(100000);

if (argc == 2) {

number\_We\_Looking\_For = atoi(argv[1]);

}

for (int i=1; i <= number\_We\_Looking\_For; i++){

if (NaturalRootedNumber(i)){

std::cout << i << std::endl;

}

}

}

bool NaturalRootedNumber( int number ){

// Алгоритм перевірки буде полягати в наступному

// Піднесемо число до квадрату та видалемо його з нього.

// в такому разі нам буде відомо чи кратне воно 10

// (це число ми будемо перевіряти на етапі 2)

int base = pow(number, 2) - number;

if (base % 10 != 0){

return false;

}

// Знайдемо розряд числа з яким ми працюємо шукаємо

int i = 0;

while ( number % int(pow(10, i)) != number) {

i++;

}

// і перевіримо остачу від ділення

// різниці квадрату нашого числа і нашого числа

if ( base % int(pow(10, i)) != 0 ) {

return false;

}

return true;

}

# Результати роботи програми

Результати роботи програми подаємо у вигляді таблиці контрольних значень (табл. 1).

Таблиця 1. Таблиця контрольних значень

|  |  |  |
| --- | --- | --- |
| № тесту | Вхідні дані | Результати |
| 1. | 5 | 1  5 |
| 2. | 10 | 1  5  6 |
| 3. | 1 00 | 1  5  6  25  76 |
| 4. | 10 000 000 | 1  5  6  25  76  376  625  9376  87232  139456 |

# Висновки

Після проведення лабораторної роботи я імплементував на практичі різноматнітні куруючі конструкції мови порграмування C++. Також було удосконалено вже існуючий алгоритм роботи валідації числа умовам задачі шляхом не використання оператора піднесення до степені і заміною цієї дії на просте множення.