ЗВІТ

про виконання лабораторної роботи № < 5.5 >

*«Обчислення сум та добутків за допомогою рекурсії »*

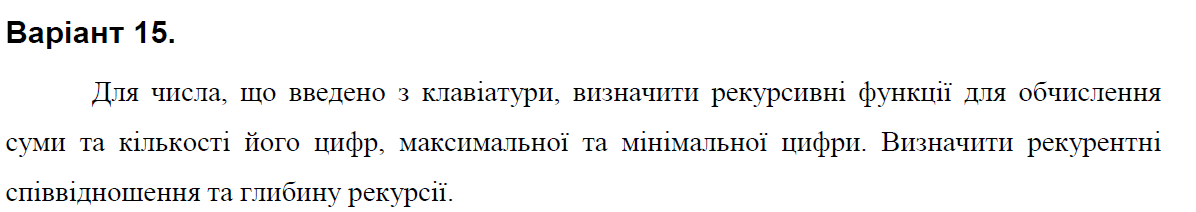
з дисципліни

«Алгоритмізація та програмування»

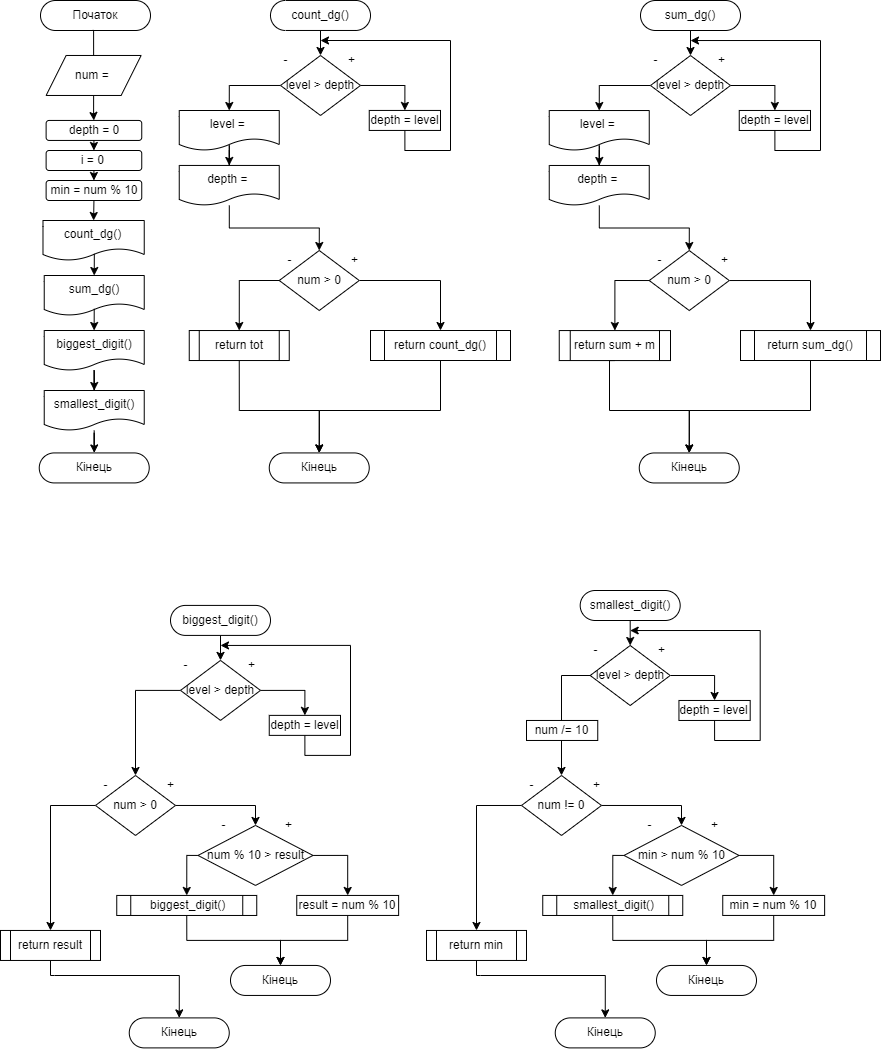
студента групи ІК-11

< *Снігура Стефана Андрійовича* >

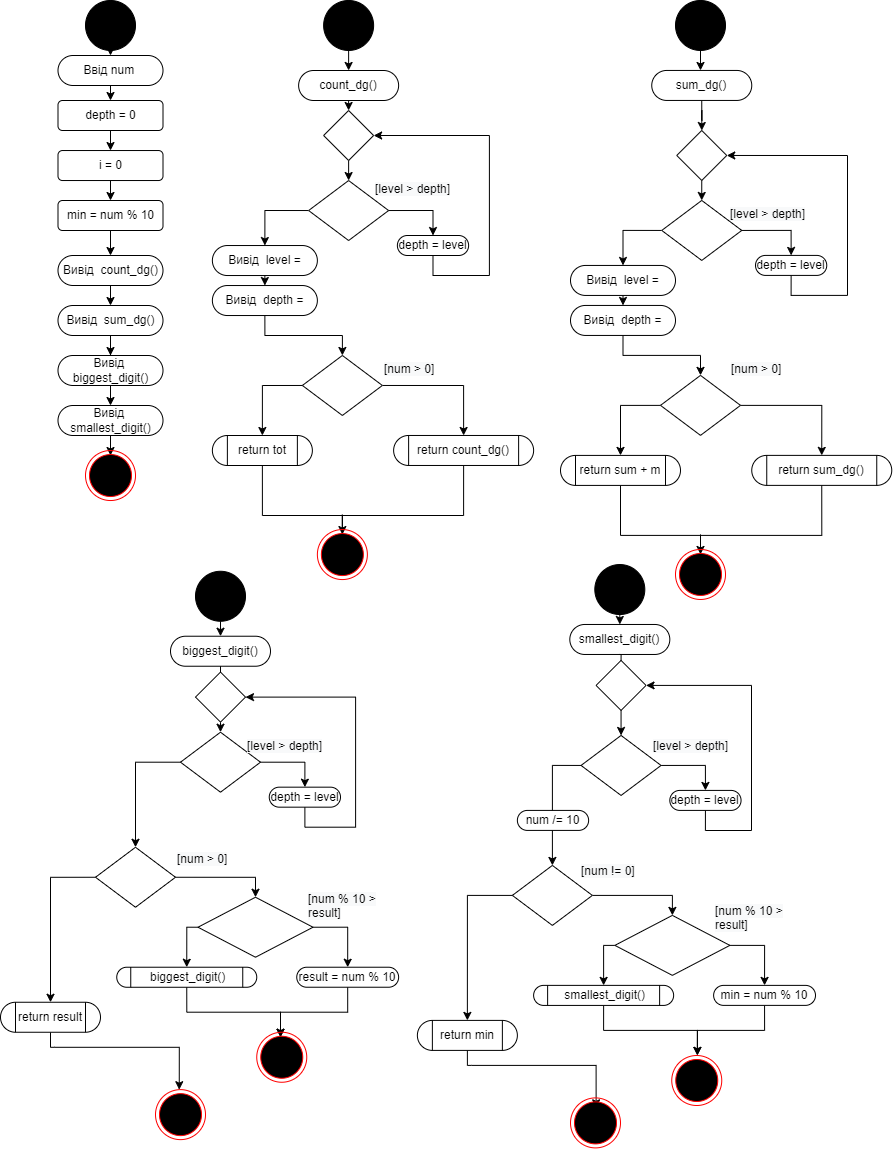
**Умова завдання:**

****

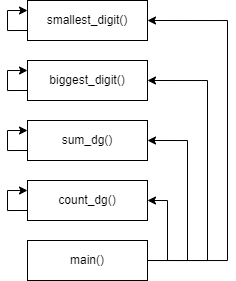
**Блок-схема:**

****

**UML-activity діаграма :**

****

**Структурна схема програми :**

****

**Текст програми:**

#include <algorithm>

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

double count\_dg(int num, int tot, int level, int depth);

double sum\_dg(int num, int m, int sum, int level, int depth);

double biggest\_digit(int num, int result, int level, int depth);

int smallest\_digit(int num, int min, int level, int depth);

int main() {

int num, m = 0, sum = 0, tot = 0;

cout << "enter numeric "; cin >> num;

int depth = 0;

int i = 0;

int min = num % 10;

cout << "Total digits = " << count\_dg(num, tot, 1, depth) << endl;

cout << "Sum of digits = " << sum\_dg(num, m, sum, 1, depth) << endl;

cout << "The biggest digit = " << biggest\_digit(num, 0, 1, depth)<< endl;

cout << "The smallest digit = " << smallest\_digit(num, min, 1, depth) << endl;

}

double count\_dg(int num, int tot, int level, int depth) {

if (level > depth) {

depth = level;

}

else

cout << "level = " << level << endl;

cout << "depth = " << depth << endl;

if (num > 0)

return count\_dg(num / 10, tot + 1, level + 1, depth);

else

return tot;

}

double sum\_dg(int num, int m, int sum, int level, int depth) {

if (level > depth) {

depth = level;

}

cout << "level = " << level << endl;

cout << "depth = " << depth << endl;

if (num > 0)

return sum\_dg(num / 10, num % 10, sum + m, level + 1, depth);

else

return (sum + m);

}

double biggest\_digit(int num, int result, int level, int depth)

{

if (level > depth) {

depth = level;

}

if(num > 0)

{

if (num % 10 > result) {

result = num % 10;

}

biggest\_digit(num / 10, result, level + 1, depth);

}

else

return result;

}

int smallest\_digit(int num, int min, int level,int depth) {

if (level > depth) {

depth = level;

}

num /= 10;

if (num != 0) {

if (min > num % 10) {

min = num % 10;

}

smallest\_digit(num, min, level + 1, depth);

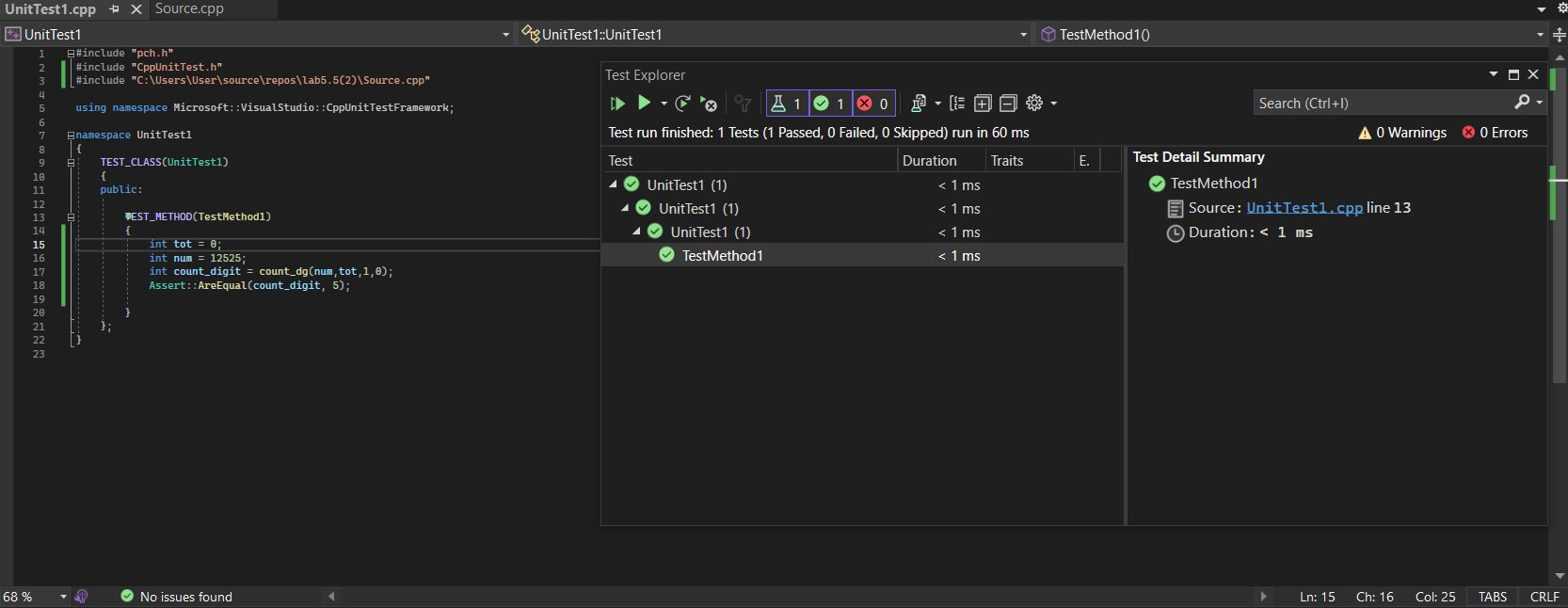
}

else

return min;

}

**UNIT-test:**

****

**Посилання на git-репозиторій з проектом:**

<https://github.com/BigTrouble-Git/ashtray.git>

**Висновки**:

Виконавши цю лабораторну роботу я навчився використовувати рекурсивні функції.