

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення



Звіт

Про виконання лабораторної роботи №1
на тему:

«Основні поняття мови С. Реалізація алгоритмів з розгалуженням в мові С»
з дисципліни «Основи програмування»

Лектор:

ст. викл. каф. ПЗ

Муха Т.О.

Виконав:

ст. гр. ПЗ-11

Ясногородський Н.В.

Прийняв:

асист. каф. ПЗ

Дивак І.В.

« __ » _____ 2021 р.

Σ = _____ .

Львів – 2021

Тема: Основні поняття мови C. Реалізація алгоритмів з розгалуженням в мові C.

Мета: Навчитися програмувати на мові C лінійні алгоритми та розгалужені обчислювальні процеси.

ЗАВДАННЯ

Завдання 1.

Написати програму для обчислення заданих арифметичних виразів. Вважати, що X, Y – змінні, значення яких слід вводити з клавіатури, всі інші величини виразу описати як константи.

$$5. \quad c = \frac{x}{a} - \frac{1}{ap} \lg(a + be^{px}), \quad \text{де: } x = 2, \quad a = 3, \quad p = 6, \quad b = 2$$

Завдання 2.

Не використовуючи оператор циклу скласти програму, яка

7. За заданим натуральним дев'ятизначним числом знаходить суму цифр цього числа та кількість одиниць.

ТЕКСТ ПРОГРАМИ

Завдання 1

```
#include <stdio.h>
#include <math.h>

const double a = 3, p = 6, b = 2;

double firstCalcTask(double x)
{
    double dangerousCalc = a + b * exp(p * x);

    if (dangerousCalc <= 0 || a == 0 || p == 0)
    {
        printf("Error: Invalid input arguments!");
        return -1;
    }
    return (x / a) - log(dangerousCalc) / (a * p);
}
```

```

int main(void)
{
    printf("Task 5 from 1st section\n");

    printf("Enter x: ");
    double x;
    scanf("%lf", &x);

    printf("Result: %lf\n\n", firstCalcTask(x));

    return 0;
}

```

Завдання 2

```

#include <stdio.h>
#include <math.h>

int intPow(int x, int y)
{
    // adding 0.5 so all compilers would round number properly
    return (int)(pow(x, y) + 0.5);
}

// find 9-digit positive integer's figures sum and "1" count
// loops are forbidden
void secondCalcTask(int nineDigitNum, int *const figuresSum, int *const oneCount)
{
    if (nineDigitNum >= intPow(10, 9) || nineDigitNum < intPow(10, 8))
    {
        printf("Error: Input number should be positive and 9-digit\n\n");
        return;
    }

    *figuresSum = getNthDigit(nineDigitNum, 9) + getNthDigit(nineDigitNum, 8) +
getNthDigit(nineDigitNum, 7) +
        getNthDigit(nineDigitNum, 6) + getNthDigit(nineDigitNum, 5) +
getNthDigit(nineDigitNum, 4) +
        getNthDigit(nineDigitNum, 3) + getNthDigit(nineDigitNum, 2) +
getNthDigit(nineDigitNum, 1);

    *oneCount = (getNthDigit(nineDigitNum, 9) == 1) + (getNthDigit(nineDigitNum, 8) == 1) +
(getNthDigit(nineDigitNum, 7) == 1) +
        (getNthDigit(nineDigitNum, 6) == 1) + (getNthDigit(nineDigitNum, 5) == 1) +
(getNthDigit(nineDigitNum, 4) == 1) +

```

```

        (getNthDigit(nineDigitNum, 3) == 1) + (getNthDigit(nineDigitNum, 2) == 1) +
        (getNthDigit(nineDigitNum, 1) == 1);
    }

    int getNthDigit(int number, int digitIdx)
    {
        return number / intPow(10, digitIdx - 1) % 10;
    }

    int main(void)
    {
        printf("Task 7 from 2nd section\n");

        int INPUT_9_DIGIT_NUMBER;
        printf("Enter 9-digit num: ");
        scanf("%d", &INPUT_9_DIGIT_NUMBER);

        int figuresSum = 0, oneCount = 0;

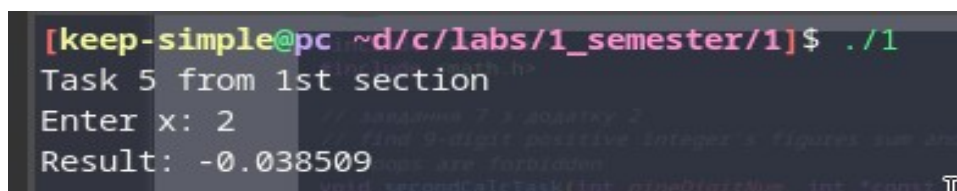
        secondCalcTask(INPUT_9_DIGIT_NUMBER, &figuresSum, &oneCount);

        printf("Figures sum: %d\n'1' count: %d\n", figuresSum, oneCount);

        return 0;
    }

```

РЕЗУЛЬТАТИ

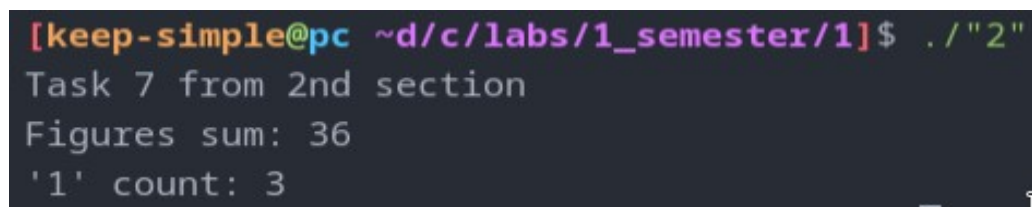


```

[keep-simple@pc ~d/c/labs/1_semester/1]$ ./1
Task 5 from 1st section
Enter x: 2
Result: -0.038509

```

Рис 1. Результат виконання програми №1



```

[keep-simple@pc ~d/c/labs/1_semester/1]$ ./"2"
Task 7 from 2nd section
Figures sum: 36
'1' count: 3

```

Рис 2. Результат виконання програми №2

ВИСНОВКИ

Виконуючи лабораторну роботу №1, я навчився(-лась) програмувати на мові С найпростіші лінійні алгоритми та алгоритми з галуженням.