

Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

Алгоритми та структури даних  
Лабораторна робота №1  
“Розгалужені оператори”

Виконав:  
студент групи ІВ-71  
Поляков М.С.  
Залікова книжка №7114  
Перевірів Сергієнко

Київ  
2017р.

**Тема:** “Розгалужені алгоритми”

**Мета:** засвоїти теоретичний матеріал та набути практичних навичок використання керуючих конструкцій розгалуження та булевих (логічних) операцій.

**Загальне завдання:**

1. Задано дійсне число  $x$ . Визначити значення заданої за варіантом кусочно-безперервної функції  $y(x)$ , якщо вона існує, або вивести на екран повідомлення про неіснування функції для даного  $x$ .
2. Розв’язати задачу двома способами (написати дві програми):
  1. В програмі дозволяється використовувати тільки одиничні операції порівняння та не дозволяється використовувати булеві (логічні) операції;
  2. В програмі необхідно обов’язково використати булеві (логічні) операції; використання булевих операцій не повинно бути надлишковим.

### ФОРМУЛЬОВКА ЗАВДАННЯ

$$y = \begin{cases} 4x^2 + 2, & x \in (-15; 3], \\ \frac{3x^3}{4} - 5, & x \in (-\infty; -30] \cup (20, +\infty) \end{cases}$$

### ХІД РОБОТИ

Текст програми I:

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

```
double y(double x) {
    // (1)
    if (x > -15) {
        if (x <= 3) {
            return 4.0 * x * x + 2.0;
        }
    }

    // (2)
    if (x <= -30) {
        return 3.0 * (x * x * x) / 4.0 - 5.0;
    }
    if (x > 20) {
        return 3.0 * (x * x * x) / 4.0 - 5.0;
    }

    return NAN;
}
```

```

int main(int argc, char **argv) {
    double x;

    // (3)
    if (argc > 1) {
        x = atof(argv[1]);
        printf("x = %lf\n", x);
    } else {
        printf("x = ");
        scanf("%lf", &x);
    }

    // (4)
    double r = y(x);

    // (5)
    if (isnan(r)) {
        printf("Invalid value entered, computation
failure.\n");
        // (5*)
        return 1;
    } else {
        printf("y(x) = %lf\n", r);
    }

    // (6)
    return 0;
}

```

Текст программы II:

```

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

```

```

double y(double x) {
    if (x > -15 && x <= 3) {
        // (1)
        return 4.0 * x * x + 2.0;
    } else if (x <= -30 || x > 20) {
        // (2)

```

```

    return 3.0 * (x * x * x) / 4.0 - 5.0;
}
return NAN;
}

int main(int argc, char **argv) {
    double x;

    // (3)
    if (argc > 1) {
        x = atof(argv[1]);
        printf("x = %lf\n", x);
    } else {
        printf("x = ");
        scanf("%lf", &x);
    }

    // (4)
    double r = y(x);

    // (5)
    if (isnan(r)) {
        printf("Invalid value entered, computation
failure.\n");
        // (5*)
        return 1;
    } else {
        printf("y(x) = %lf\n", r);
    }

    // (6)
    return 0;
}

```

### **ПОЯСНЕННЯ ДО РОЗВ'ЯЗКІВ**

Функція  $y(x)$  вертає NaN (не число), якщо число  $x$  не попадає до жодної з умов рядків.

1. Перший рядок кусочно-безперервної функції;
2. Другий рядок кусочно-безперервної функції;
3. Якщо число задано аргументом командної строки, то використовувати його, інакше запросити дані в користувача;

4. Обчислити значення функції  $y(x)$  та зберегти результат в  $g$ ;
5. Якщо  $g$  означає “не число”, то вивести помилку, інакше вивести результат;  
(\*) Вийти з кодом 1 (помилка);
6. Вийти з кодом 0 (успішне виконання).

**ТЕСТОВІ ДАНІ**

х	Вивід програми І	Вивід програми ІІ
1.0	6.0	6.0
-17.0	Помилка обчислень	Помилка обчислень
-32.0	-24581.0	-24581.0
3.0	38.0	38.0
17.0	Помилка обчислень	Помилка обчислень
21.0	6940.75	6940.75