

**Міністерство освіти і науки  
України Львівський національний університет імені Івана Франка  
Факультет електроніки та комп'ютерних технологій**

**Звіт**

**про виконання лабораторної роботи №5**

**“УМОВНІ ОПЕРАТОРИ”**

Виконав:

студент 1 курсу

групи ФЕП-11с

Абдулханов Абдул-Рахім

Викладач:

ас. Кужій.Ю.І.

**Львів - 2024**

github з кодом:

Завдання 2:

```
#include <stdio.h>

int main(void){
    int a, b;
    printf("First num:");
    scanf("%d", &a);
    printf("Second num:");
    scanf("%d", &b);
    a > b ? printf("first number is bigger than second\n")
    : a < b ? printf("second number is bigger than first\n")
    : printf("first number is equal to the second\n");
    return 0;
}
```

```
First num:4
Second num:2
first number is bigger than second
```

Завдання 3:

```
#include <stdio.h>

int main(void){
    int a, b, c, temp;
    printf("First side:");
    scanf("%d", &a);
    printf("Second side:");
    scanf("%d", &b);
    printf("Third side:");
    scanf("%d", &c);

    if(a + b > c && a + c > b && b + c > a){
        if(a == b && b == c){
            printf("трикутник рівносторонній\n");
        }
        else if(a == b || a == c || b == c){
            printf("трикутник рівнобедренний\n");
        }
        else{
            printf("трикутник різносторонній\n");
        }

        if(a >= b && a >= c) {
            temp = a;
            a = c;
            c = temp;
        }
        else if(b >= a && b >= c){
            temp = b;
            b = c;
            c = temp;
        }

        if(c*c == a*a + b*b) printf("трикутник прямокутний\n");
        else if (c*c < a*a + b*b) printf("трикутник гострокутний\n");
        else printf("трикутник тупокутний\n");
    }
    else{
        printf("такого трикутника не існує\n");
    }
    return 0;
}
```

```
First side:3
Second side:4
Third side:5
трикутник різносторонній
трикутник прямокутний
```

#### Завдання 4:

```
#include <stdio.h>

void int_to_string(int a){
    char *hundreds[] = {"", "сто", "двісті", "триста", "чотириста", "п'ятсот", "шістсот", "сімсот", "вісімсот", "дев'ятсот"};
    char *tens[] = {"", "", "двадцять", "тридцять", "сорок", "п'ятдесят", "шістдесят", "сімдесят", "вісімдесят", "дев'яносто"};
    char *teens[] = {"десять", "одинадцять", "дванадцять", "тринадцять", "чотирнадцять", "п'ятнадцять", "шістнадцять", "сімнадцять", "вісімнадцять", "дев'ятнадцять"};
    char *units[] = {"", "один", "два", "три", "чотири", "п'ять", "шість", "сім", "вісім", "дев'ять"};

    printf("%s ", hundreds[a / 100]);
    int remainder = a % 100;

    if(remainder >= 10 && remainder < 20){
        printf("%s", teens[remainder - 10]);
    }
    else{
        printf("%s %s\n", tens[remainder / 10], units[remainder % 10]);
    }
}

int main(void){
    int a;
    printf("введіть тризначне число:");
    scanf("%d", &a);

    if(a < 100 || a >= 1000){
        printf("треба тризначне\n");
        return 0;
    }
    else{
        int_to_string(a);
    }

    return 0;
}
```

```
введіть тризначне число:352
триста п'ятдесят два
```

Висновок: У цій лабораторній роботі я ознайомився з умовними операторами в мові C, вивчаючи, як використовувати оператори для виконання різних логічних умов. Розробка програм для порівняння чисел за допомогою тернарного оператора, визначення типу трикутника за заданими сторонами, а також виведення тризначного числа прописом продемонструвала практичне застосування умовних конструкцій. Вивчаючи ці концепції, я зрозумів важливість умовних операторів для управління потоком виконання програми. Вони дозволяють реалізувати складну логіку прийняття рішень, що є критично важливим для розробки програмного забезпечення. Загалом, вивчення умовних операторів є

ключовим етапом у вивченні програмування, оскільки вони формують основи для розуміння більш складних концепцій, таких як цикли та функції.