

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
УПРАВЛІННЯ

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №3
ЗА ТЕМОЮ «ВИКОРИСТАННЯ ФУНКЦІЙ»

Виконав студент
групи КН-221в
Шулюпов Єгор Русланович

Перевірів
Асистент кафедри
Ольховий Олексій Михайлович

ХАРКІВ 2021

Мета

- 1) Навчитися реалізовувати різні види алгоритмів на мові програмування C++ за допомогою функцій
- 2) Закріпити здобуті навички з основ програмування на C++ з використанням підпрограм, що викликаються

Хід роботи

Завдання 1.1 Статичні локальні змінні

Написати програму, яка обчислює та виводить мінімум і максимум цілих чисел, в міру того, як користувач вводить ці числа. Слід використати статичні локальні змінні.

```
#include <iostream>

using namespace std;

int main()
{
    static int max = INT_MIN; static int min = INT_MAX; int a; bool
trigger;
    while (true)
    {
        cin >> a;
        if (a > max) {
            max = a;
            if (a < min) {
                min = a;
            }
        }
        else {
            if (a < min) {
                min = a;
            }
        }
        cout << "min = " << min << endl << "max = " << max << endl
<< endl;
        cout << "print (1) to continue" << endl;
        cout << "print (0) to exit" << endl;
        cin >> trigger;
        if (trigger == true) {
            continue;
        }
        else {
            break;
        }
    }
    return 0;
}
```

Тести

```
11
min = 11
max = 11

print (1) to continue
print (0) to exit
1
-5
min = -5
max = 11

print (1) to continue
print (0) to exit
1
4
min = -5
max = 11

print (1) to continue
print (0) to exit
1
12
min = -5
max = 12

print (1) to continue
print (0) to exit
1
0
min = -5
max = 12

print (1) to continue
print (0) to exit
-
```

Завдання 1.2 Рекурсія

Написати програму, яка зчитує x і n і обчислює y за допомогою рекурсивної функції:

$$y = (x + 1)(x + 2)(x + 3)(x + 4) \dots (x + n)$$

```
#include <iostream>

using namespace std;

int product(int i, int n, double x, int y) {
    y *= (x + i);
    i++;
    if (i <= n) {
        product(i, n, x, y);
    }
    else cout << "y = " << y << endl;
    return y;
}

int main () {
    int i = 1; double x; int n; int y = 1;

    cout << "input x = "; cin >> x;
    cout << "input n = "; cin >> n;
    if (n > 0) {
        cout << product(i, n, x, y);
    }
}
```

```

    }
    else {
        cout << "data must be n > 0";
    }
    return 0;
}

```

Тести

```

input x = 1
input n = 4
y = 120

```

Завдання 1.3 Аргументи з усталеними значеннями

Створити функції, які повертають 1, аргумент, і добуток аргументів, залежно від кількості аргументів. Першу функцію реалізувати з усталеними значеннями аргументів, інші функції – через механізм перевантаження імен. У функції main() слід здійснити тестування всіх функцій.

```

#include <iostream>

using namespace std;

double qwe() {
    return 1;
}

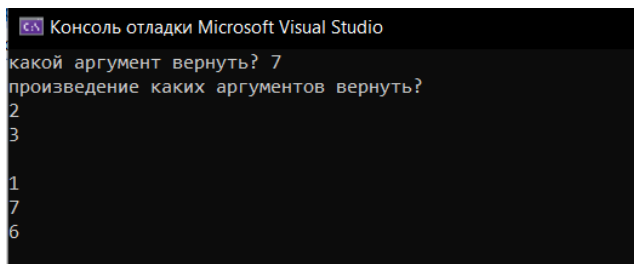
double qwe(double x) {
    return x;
}

double qwe(double a, double b) {
    return a * b;
}

int main() {
    setlocale(LC_ALL, "Ukrainian");
    double x, a, b;
    cout << "какой аргумент вернуть? "; cin >> x;
    cout << "произведение каких аргументов вернуть? "; cin >> a >> b; cout << endl;
    cout << qwe() << endl;
    cout << qwe(x) << endl;
    cout << qwe(a, b) << endl;
}

```

```
} }
```



```
Консоль отладки Microsoft Visual Studio
какой аргумент вернуть? 7
произведение каких аргументов вернуть?
2
3
1
7
6
```

Завдання 1.4 Квадратне рівняння

Створити функцію для розв'язання квадратного рівняння. Функція повинна повертати кількість коренів (0, 1 або 2) або -1, якщо рівняння має безліч розв'язків. Функція повинна отримати коефіцієнти як аргументи та повертати корені як аргументи-посилання.

```
#include <iostream>
#include <cmath>
using namespace std;
int sq_eq(double a, double b, double c)
{
    if (a != 0) {
        double d = pow(b, 2) - 4 * a * c;
        if (d >= 0) {
            if (d > 0) {
                cout << (-b + sqrt(d)) / (2 * a) << endl;
                cout << (-b - sqrt(d)) / (2 * a) << endl;
                return 2;
            }
            else {
                cout << -b / (2 * a) << endl;
                return 1;
            }
        }
        else {
            cout << "no roots" << endl;
            return 0;
        }
    }
    else {
        if (b != 0) {
            cout << -c / b << endl;
            return 1;
        }
        else {
            if (c != 0) {
                cout << "no roots" << endl;
                return 0;
            }
            else {
                cout << "any value" << endl;
                return -1;
            }
        }
    }
}
```

```
int main() {
    setlocale(LC_ALL, "UKRAINIAN");
    double a, b, c; cin >> a >> b >> c;
    cout << endl;
    cout << "кількість коренів = " << sq_eq(a, b, c) << endl;
    return 0;}

```

```

1
1
-2

1
-2
к?лк?сть корен?в = 2
1
1
2

no roots
к?лк?сть корен?в = 0
000
0
0

any value
к?лк?сть корен?в = -1
0
1
2

-2
к?лк?сть корен?в = 1
-

```

Завдання 1.5 Індивідуальне завдання

11, 27

$$y = \begin{cases} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{1}{x-i+xj}, x \leq 0 \\ \prod_{i=1}^n \left(\frac{1}{x} - \frac{1}{i} \right), x > 0 \end{cases}$$

```

#include <iostream>

using namespace std;

double input_x_2() {
    double x_2;
    cout << "x_2 : " << endl;
    cin >> x_2;
    return x_2;
}

double input_step() {
    double step;
    cout << "step : " << endl;
    cin >> step;
    return step;
}

int y(double x_1, int n)
{
    double step = input_step();

```

```

double x_2 = input_x_2();
if ((n < 1) || (step <= 0) || (x_1 > x_2)) {
    cout << "data must be ((n >= 1) || (step > 0) || (x_1 <= x_2))" << endl;
    return y(x_1, n);
}

while (x_1 <= x_2) {
    if (x_1 <= 0) {
        double y = 0; int i = 0;
        if (x_1 != 0) {
            for (i; i <= (n - 1); i++) {
                int j = 0;
                for (j; j <= (n - 1); j++) {
                    y += 1 / (x_1 - i + x_1 * j);
                }
            }
            cout << "(x ; y) : " << x_1 << " ; " << y << endl;
        }
        else {
            cout << "division by zero (x) " << x_1 << endl;
        }
    }
    else {
        double y = 1; int i = 1;
        for (i; i <= n; i++) {
            y *= (1 / x_1 - 1 / i);
        }
        cout << "(x ; y) : " << "(" << x_1 << " ; " << y << ")" << endl;
    }
    x_1 += step;
}
return 0;
}

int main() {
    double x_1; int n;
    cout << "x_1 : "; cin >> x_1;
    cout << "n : "; cin >> n;
    y(x_1, n);
    return 0;
}

```

```

x_1 : -5
n : 2
step :
1
x_2 :
5
(x ; y) : -5 ; -0.557576
(x ; y) : -4 ; -0.686111
(x ; y) : -3 ; -0.892857
(x ; y) : -2 ; -1.28333
(x ; y) : -1 ; -2.33333
division by zero (x) 0
(x ; y) : (1 ; 0)
(x ; y) : (2 ; -0.25)
(x ; y) : (3 ; -0.222222)
(x ; y) : (4 ; -0.1875)
(x ; y) : (5 ; -0.16)

C:\Users\esulu\Desktop\folders\1 курс\labs_op\лаб_3\Project1_5\x64\Debug\Project5.
кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "С

```

Вправи для контролю

Завдання 1

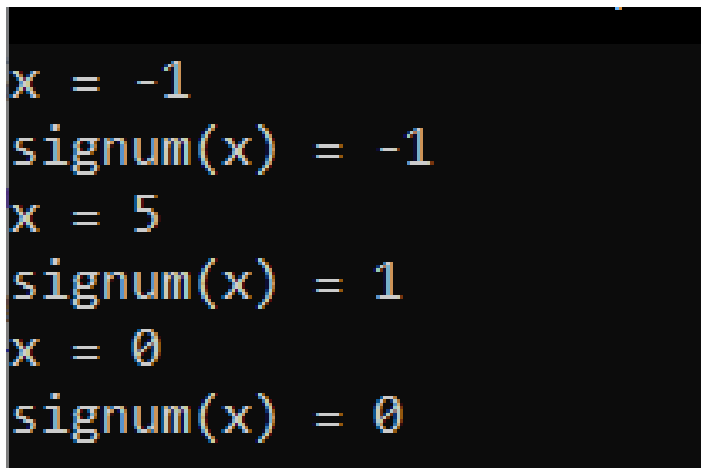
Створити програму для тестування функції `signum()`.

```
#include <iostream>

using namespace std;

int signum(double x) {
    int y;
    if (x > 0) {
        y = 1;
        return y;
    };
    if (x < 0) {
        y = -1;
        return y;
    };
    if (x == 0) {
        y = 0;
        return y;
    };
}

int main() {
    double x;
    cout << "x = ";
    cin >> x;
    cout << "signum(x) = " << signum(x) << endl;
    return 0;
}
```



```
x = -1
signum(x) = -1
x = 5
signum(x) = 1
x = 0
signum(x) = 0
```

Завдання 2

Здійснити розробку та тестування функції, яка обчислює добуток трьох аргументів.

```
#include <iostream>
```



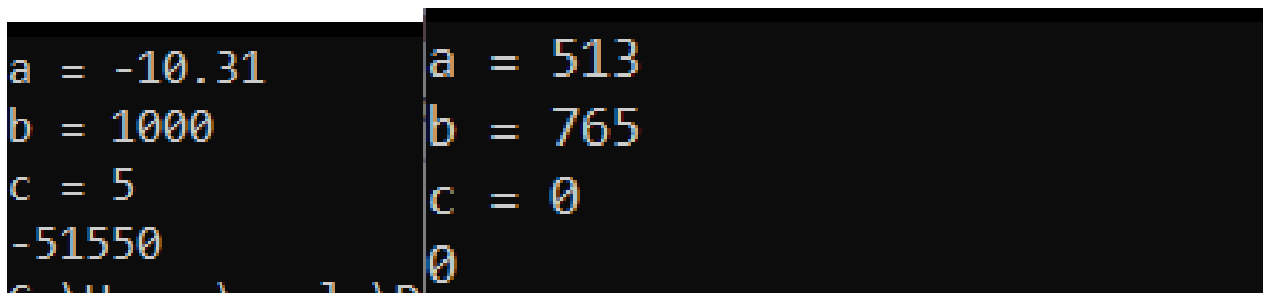
```

using namespace std;

double multiply(double a, double b, double c) {
    return a * b * c;
}

int main() {
    double a, b, c;
    cout << "a = "; cin >> a;
    cout << "b = "; cin >> b;
    cout << "c = "; cin >> c;
    cout << multiply(a, b, c);
    return 0;
}

```



```

a = -10.31
b = 1000
c = 5
-51550

a = 513
b = 765
c = 0
0

```

Завдання 3

Здійснити розробку та тестування функції, яка обчислює добуток перших n непарних значень

```

#include <iostream>

using namespace std;

double multiply(int n) {
    int y = 1, a = 1;
    for (int i = 1; i <= n; i++) {
        y *= a;
        a += 2;
    }
    return y;
}

int main() {
    int n;
    cout << "input n = "; cin >> n;
    if (n > 1) {
        cout << "y = " << multiply(n);
    }
    else
        cout << "data must be n > 0";
    return 0;
}

```

```
input n = -5  
data must be n > 0
```

```
input n = 3  
y = 15
```

Завдання 4

Здійснити розробку та тестування функції, яка обчислює e^x через суму

```
#include <iostream>  
#include <cmath>  
using namespace std;  
  
double ePowX(double x)  
{  
    double z = 1, y = 0, i = 1;  
  
    while (z > 0.000000000000000001)  
    {  
        y += z;  
        z *= x / i;  
        i++;  
    }  
    return round(y * 10000) / 10000;  
}  
  
int main()  
{  
    setlocale(LC_ALL, "ru");  
  
    double x;  
  
    cout << "Введите степень, в которую хотите возвести e: ";  
    cin >> x;  
  
    cout << "e ^ " << x << " = " << ePowX(x)<< endl;  
}
```

```
Введите степень, в которую хотите возвести e: 2  
e ^ 2 = 7.3891
```

Завдання 5

Здійснити розробку та тестування функції, яка обчислює факторіал

```
#include <iostream>  
  
using namespace std;
```

```

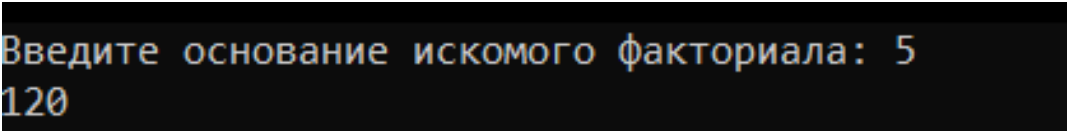
int factorial(int n) {
    int y = 0;
    if (n == 0) {
        y = 1;
    } else
    if (n < 0) {
        cout << "data must be n >= 0 ";
    } else
    if (n > 0) {
        y = 1;
        for (int i = 1; i <= n; i++) {
            y *= i;
        }
    }
    return y;
}

int main() {
    setlocale(LC_ALL, "Russian");

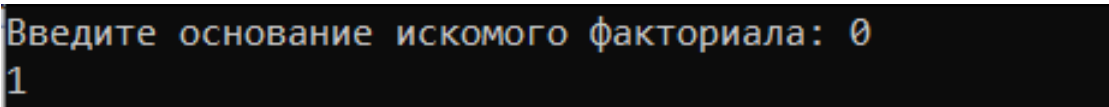
    int n;
    cout << "Введите основание искомого факториала: "; cin >> n;
    cout << factorial(n);

    return 0;
}

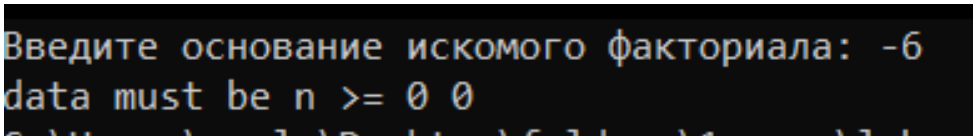
```



Введите основание искомого факториала: 5
120



Введите основание искомого факториала: 0
1



Введите основание искомого факториала: -6
data must be n >= 0 0

Завдання 6

Здійснити розробку та тестування функції, яка здійснює виведення всіх парних значень в заданому діапазоні

```

#include <iostream>
using namespace std;

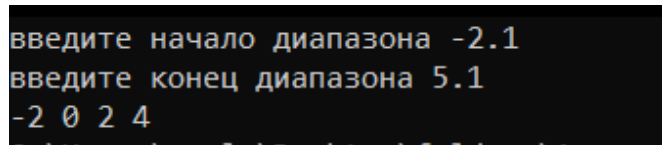
```

```

int even(double x_1, double x_2) {
    int x = ceil(x_1);
    for (x; x <= x_2; x++) {
        if ((x % 2) == 0) {
            cout << x << " ";
        }
    }
    return 0;
}

int main() {
    setlocale(LC_ALL, "Russian");
    double x_1, x_2;
    cout << "введите начало диапазона "; cin >> x_1;
    cout << "введите конец диапазона "; cin >> x_2;
    even(x_1, x_2);
    return 0;
}

```



```

введите начало диапазона -2.1
введите конец диапазона 5.1
-2 0 2 4

```

Завдання 7

Здійснити розробку та тестування функції, яка здійснює виведення добутку перших n парних значень

```

#include <iostream>

using namespace std;

double multiply(int n) {
    int y = 1, a = 2;
    for (int i = 1; i <= n; i++) {
        y *= a;
        a += 2;
    }
    return y;
}

int main() {
    int n;
    cout << "input n = "; cin >> n;
    if (n > 1) {
        cout << "y = " << multiply(n);
    }
    else
        cout << "data must be n > 0";
    return 0;
}

```

```
input n = 3  
y = 48
```

```
input n = -5  
data must be n > 0
```

Завдання 8

Здійснити розробку та тестування функції, яка обчислює найбільший спільний дільник двох цілих чисел

```
#include <iostream>  
  
using namespace std;  
  
int gcd(int a, int b) {  
    while (a > 0 && b > 0) {  
        if (a > b) {  
            a %= b;  
        }  
        else {  
            b %= a;  
        }  
    }  
  
    return a + b;  
}  
  
int main() {  
    setlocale(LC_ALL, "Russian");  
    int a, b;  
    cout << "a = "; cin >> a;  
    cout << "b = "; cin >> b;  
    cout << "НОД: " << gcd(a, b) << endl;  
    return 0;  
}
```

```
a = 63  
b = 27  
НОД: 9
```

```
a = 144  
b = 12  
НОД: 12
```

Висновок:

Отже, в ході роботи я навчився реалізовувати різні види алгоритмів на мові програмування C++ за допомогою функцій, закріпив здобуті навички з основ програмування на цій мові, використовуючи підпрограми.