

Задача 1 (4 точки). Запишете на езика C++ следните математически формули:

а) $\frac{a^2 + b^2}{\sqrt{c^4 + b^4}} \cdot \frac{d}{e+f}$

`((a*a + b*b)/sqrt(pow(c, 4)+pow(b, 4))) * (d*(e+f))`

б) $\frac{\sqrt{2} + \sqrt[3]{3} + \sqrt[4]{4}}{\sqrt[5]{5} + \sqrt[6]{6} + \sqrt[7]{7}}$

`(sqrt(2) + pow(3, (double)1 / 3) + pow(4, (double)1/4))/(pow(5, (double)1/5) + pow(6, (double)1/6) + pow(7, (double)1/7))`

в) $\frac{(\sin x^2 + \cos x^3)^2}{\lg|2 + x|}$

`pow(sin(x*x) + cos(x*x*x), 2) / log10(abs(2+x))`

г) $\frac{(\log_2 |x| + e^{\frac{x+y}{2}})^3}{\lg(2 + e^{\frac{x-y}{2}})}$

`pow(log(abs(x))/log2 + exp((x+y)/2), 3) / log10(2+ exp((x-y)/2))`

Задача 2 (4 точки). Да се напише БУЛЕВ ИЗРАЗ, който има стойност истина, ако посоченото условие е вярно и стойност – лъжа, в противен случай:

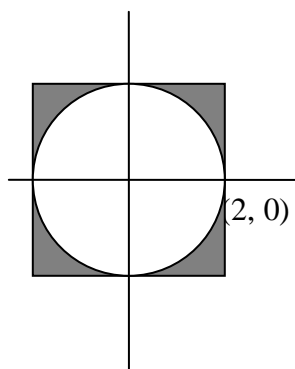
а) поне едно от числата a, b и c е положително

`bool expression = (a > 0 || b > 0 || c > 0);`

б) x е равно на $\min\{a, b, c\}$

`bool expression = ((x == a && a < b && a < c) || (x == b && b < c && b < a) || (x == c && c < b && c < a));`

в) точка принадлежи на заштрихованата част на фигурата



```
double x, y;
bool expr = (!(x*x+y*y <= 2*2) && abs(x) <= 2 && abs(y) <= 2);
д) цифрата 5 влиза в запис на положителното четирицифрено число n.
bool expr = (n/1000 == 5 || n/100%10 == 5 || n/10%10 == 5 || n%10 == 5);
```

Задача 3 (4 точки). Да се напише условен оператор, който е еквивалентен на оператора за присвояване `x = !a || b && !c`; където всички променливи са булеви и в който не се използват логически операции.

```
bool x;
if(a) {
    if(c) {
        x = false;
    } else {
        if(b) {
            x = true;
        } else {
            x = false;
        }
    }
} else {
    x = true;
}
```

Задача 4 (4 точки). Условните оператори, използвани в задача 3, да се запишат чрез еквивалентни switch оператори.

```
switch(a) {
    case true:
        switch(c) {
            case true: x = false;
            default:
                switch(b) {
                    case true: x = true;
                    default: x = false;
                }
        }
    default: x = true;
}
```

Задача 5 (4 точки). Да се напише програма, която намира стойността на верижната дроб

$$1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\dots + \frac{1}{109 + \frac{1}{111}}}}}$$

```
double result = 111;
    for(size_t i = 109; i >= 1; i-=2) {
        double helper = 1/result;
        result = i + helper;
    }
```

Задача 6 (4 точки). Да се напише програма, която намира стойността на $\ln(x)$ по формулата:

$$\ln(x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n+1} \frac{x^n}{n} + \dots$$

Пресмятането да продължи докато последното добавено събираемо по модул стане по-малко от ϵ .

```
double epsilon;
cin >> epsilon;
double x;
cin >> x;
double current = x;
double result = 0;
int i = 1;
while(abs(current) >= epsilon) {
    current = pow(x, i)/i;
    result += pow(-1, i+1)*current;
}
```

Задача 7 (4 точки). Едно естествено число е съвършено, ако е равно на сумата от своите делители (без самото число). Например, 6 е съвършено, защото $6 = 1+2+3$. Да се дефинира функция, която проверява дали естествено число е съвършено.

```
bool isPerfect(int n) {
    int sumOfDivisors = 0;
    for(size_t i = 1; i < n; i++) {
        if(n%i==0) {
            sumOfDivisors += i;
        }
    }
    return sumOfDivisors == n;
}
```

```
}
```

Задача 8 (4 точки). Да се напише програмен фрагмент, който проверява дали в редицата от естествени числа a_0, a_1, \dots, a_{n-1} има съвършено число. За целта да се използва функцията, дефинирана в предходната задача.

```
//нека редицата е arr, с дължина n
bool containsSuch(int* arr, int n) {
    for(int i = 0; i < n; i++) {
        if(isPrime(arr[i])) {
            return true;
        }
    }
    return false; //може и без този ред;
}
```

Задача 9 (4 точки). Какъв е резултатът от изпълнението на програмата?

```
#include <iostream.h>
const int a = 7;
void func(const int& x, const int* y = &a,
          int z = 6)
{
    int a = 4;
    int b = 5;
    z = x + *y + a + b;
    cout << "func: x = " << x << endl;
    cout << "func: *y = " << *y << endl;
    cout << "func: z = " << z << endl;
    cout << "func: a = " << a << endl;
    cout << "func: b = " << b << endl;
    return;
}
int main()
{
    int a = 1;
    int b = 2;
    int c = 3;
    func(a, &b, c);
    func(b, &c);
    func(c);
    cout << "main: a = " << a << endl;
    cout << "main: b = " << b << endl;
    cout << "main: c = " << c << endl;
    return 0;
}
```

```
func: x = 1
func: *y = 2
func: z = 12
func: a = 4
func: b = 5
func: x = 2
func: *y = 3
func: z = 14
func: a = 4
func: b = 5
func: x = 3
func: *y = 7
func: z = 19
func: a = 4
func: b = 5
main: a = 1
main: b = 2
main: c = 3
```

Задача 10 (4 точки). Да се дефинира функция, която умножава матриците $A_{n \times m}$ и $B_{m \times k}$.

```
void multiply(double** mat1, double** mat2, double** res, int n, int m, int k)
{
    int i, j, k;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            res[i][j] = 0;
            for (k = 0; k < K; k++)
                res[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}
```

Задача 11 (4 точки). Да се дефинира *рекурсивна* функция, която намира броя на срещанията на дадена цифра k в десетичния запис на дадено естествено число n .

```
int occurrences(int n, int k) {
    if(n < 10 && n != k) {
        return 0;
    }
    if(n % 10 == k) {
        return 1 + occurrences(n/10, k);
    }
    return occurrences(n/10, k);
}
```

Задача 12 (4 точки). Да се дефинира *рекурсивна* функция, която намира максималния елемент на редицата $a_0, a_1, a_2, \dots, a_{n-1}$, където $n \geq 1$.

```
int max(int a, int b) {
    return (a > b) ? a : b;
}
```

```
int maxEl(int* arr, int n) {  
    if(n == 1) {  
        return arr[0];  
    }  
    return max(arr[n-1], maxEl(arr, n-1));  
}
```