Ministry of Education of Republic of Moldova
Technical University of Moldova

Report

on Computer Programming

Laboratory Work Nr. 7

Performed by b.**Brinza Cristian FAF-212**

Verified by lect.univ **Gaidău M.**

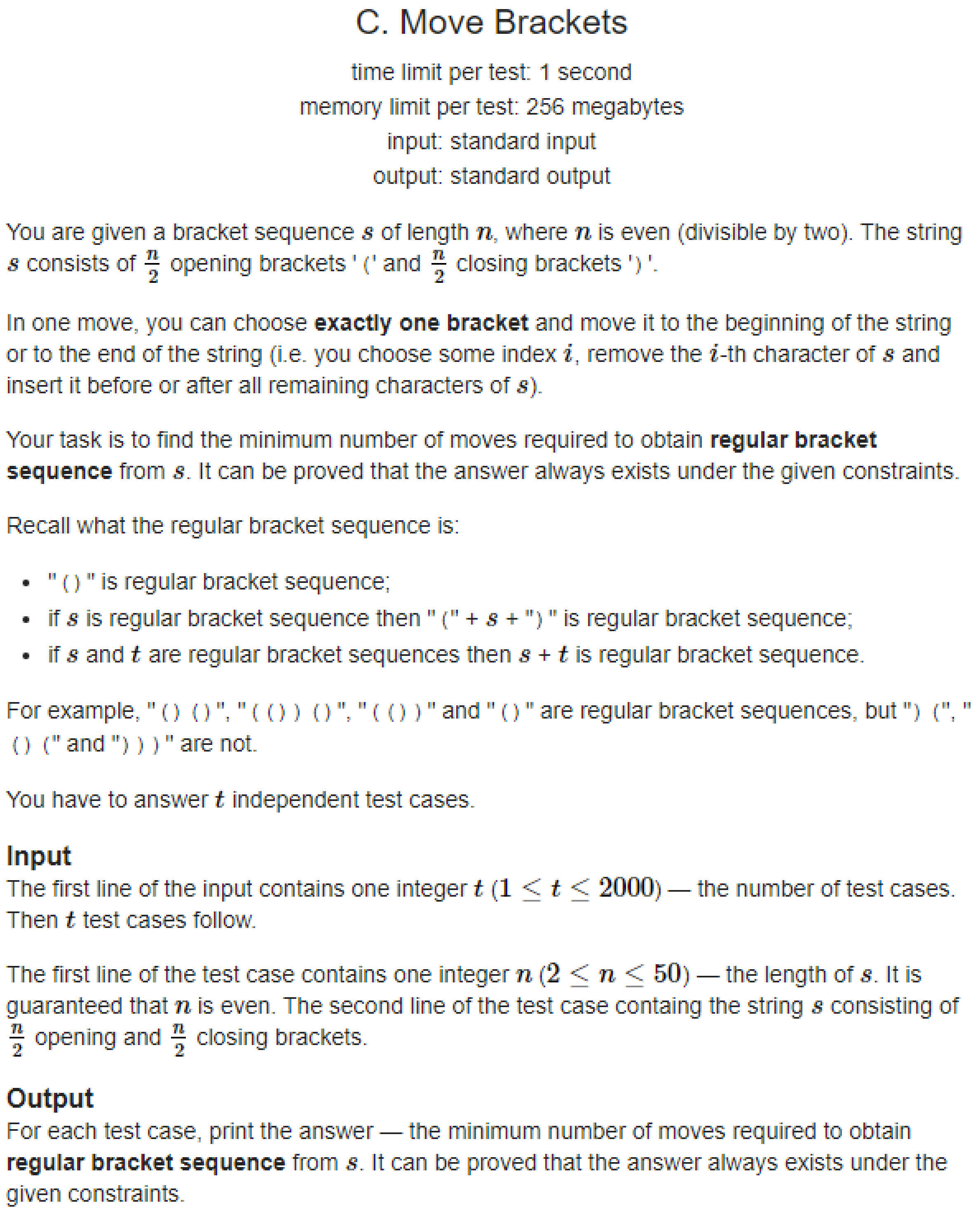
Chișinău 2021

Topic: Pointers in C/C++ languages, Pointer operators & and *, Declaring pointers

Purpose of the laboratory work: Accumulation of practical skills for developing and programming computational processes and program testing skills.

Condition of the problem:

(Codeforces. Problem 653 (Div. 3)C, https://codeforces.com/contest/1374/problem/C**)**



C. Move Brackets

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a bracket sequence s of length n, where n is even (divisible by two). The string s consists of $\frac{n}{2}$ opening brackets '(' and $\frac{n}{2}$ closing brackets ')'.

In one move, you can choose **exactly one bracket** and move it to the beginning of the string or to the end of the string (i.e. you choose some index i, remove the i-th character of s and insert it before or after all remaining characters of s).

Your task is to find the minimum number of moves required to obtain **regular bracket sequence** from s. It can be proved that the answer always exists under the given constraints.

Recall what the regular bracket sequence is:

- "()" is regular bracket sequence;
- if s is regular bracket sequence then "(" + s + ")" is regular bracket sequence;
- if s and t are regular bracket sequences then s + t is regular bracket sequence.

For example, "()()", "(())()", "(())" and "()" are regular bracket sequences, but ")(", "()(" and ")))" are not.

You have to answer t independent test cases.

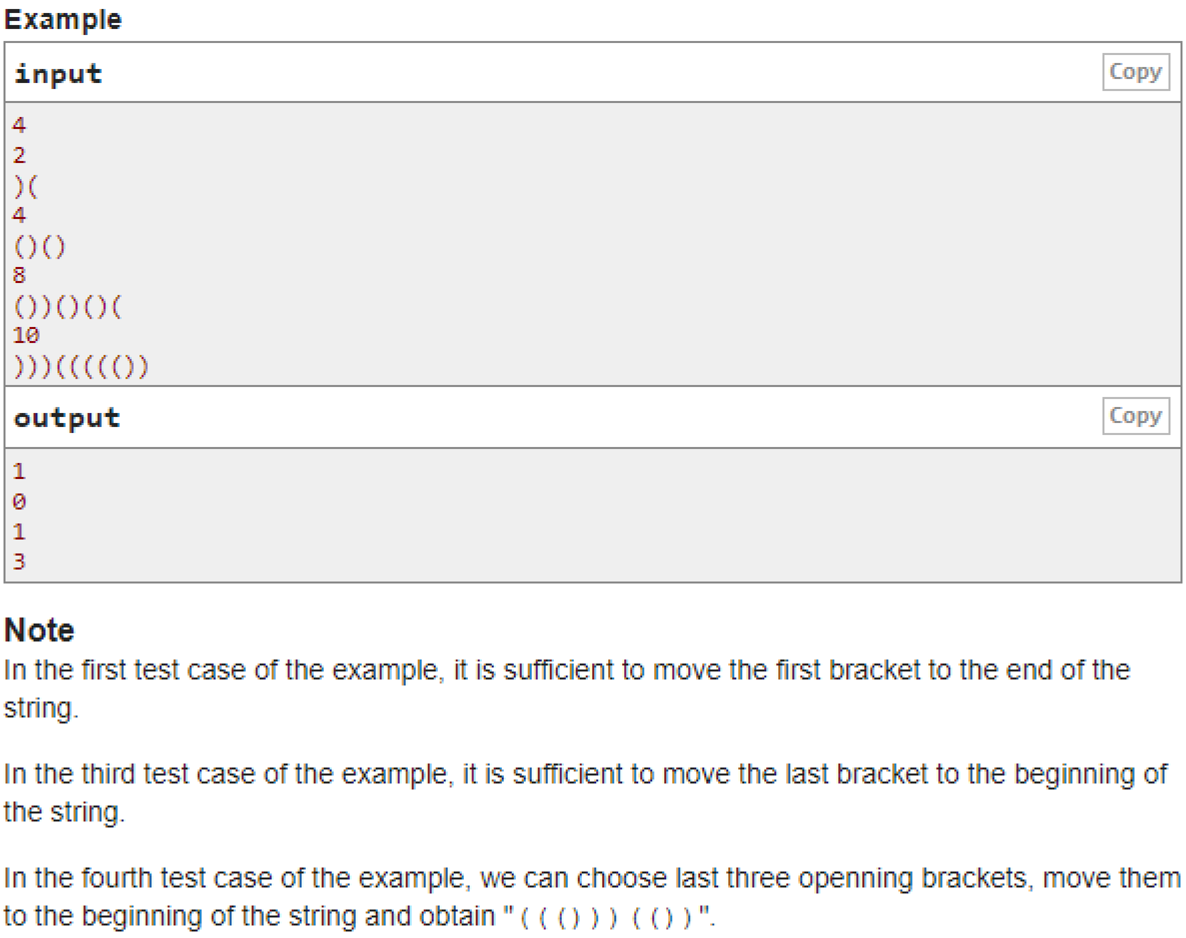
Input

The first line of the input contains one integer t ($1 \le t \le 2000$) — the number of test cases. Then t test cases follow.

The first line of the test case contains one integer n ($2 \le n \le 50$) — the length of s. It is guaranteed that n is even. The second line of the test case containg the string s consisting of $\frac{n}{2}$ opening and $\frac{n}{2}$ closing brackets.

Output

For each test case, print the answer — the minimum number of moves required to obtain **regular bracket sequence** from s. It can be proved that the answer always exists under the given constraints.



Example

input

```
4
2
)(
4
()()
8
())()()(
10
)))((((())
```

output

```
1
0
1
3
```

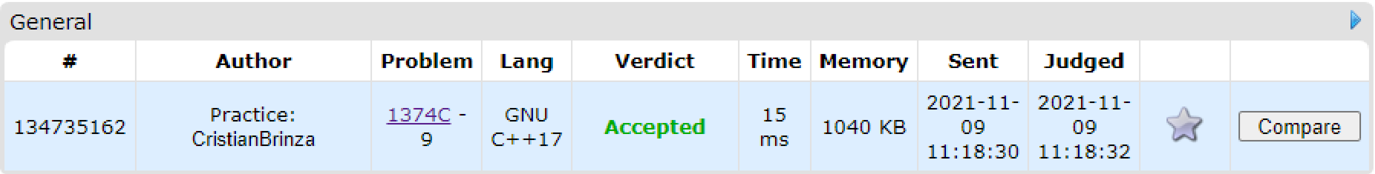
Note

In the first test case of the example, it is sufficient to move the first bracket to the end of the string.

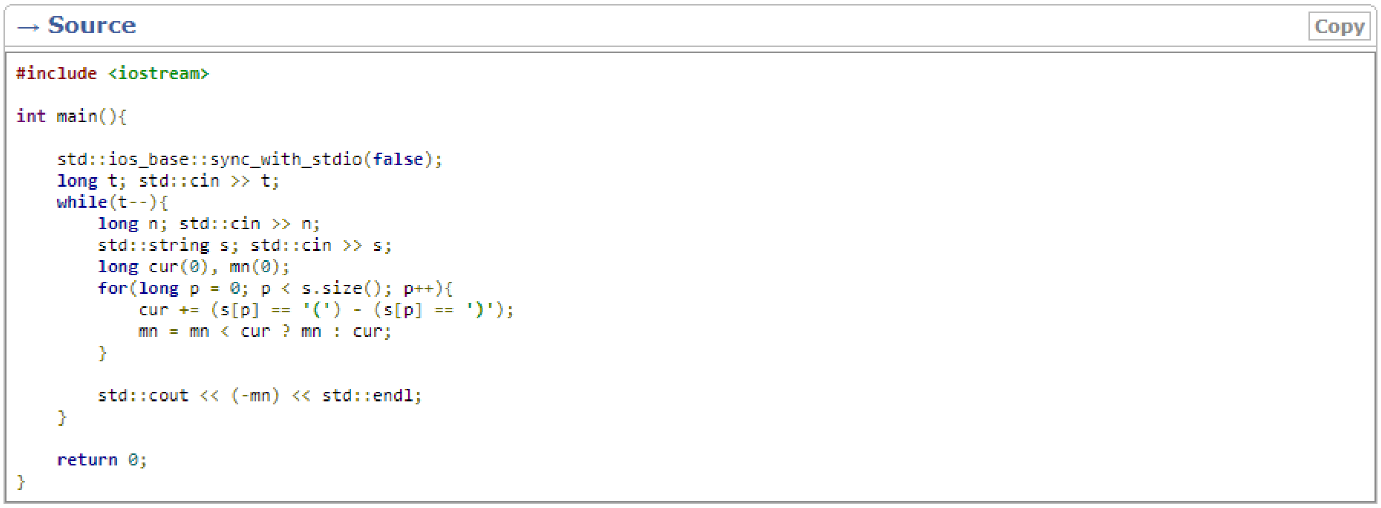
In the third test case of the example, it is sufficient to move the last bracket to the beginning of the string.

In the fourth test case of the example, we can choose last three openning brackets, move them to the beginning of the string and obtain "((()))(())".

Work processing:



#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
134735162	Practice: CristianBrinza	1374C - 9	GNU C++17	Accepted	15 ms	1040 KB	2021-11-09 11:18:30	2021-11-09 11:18:32	☆	Compare



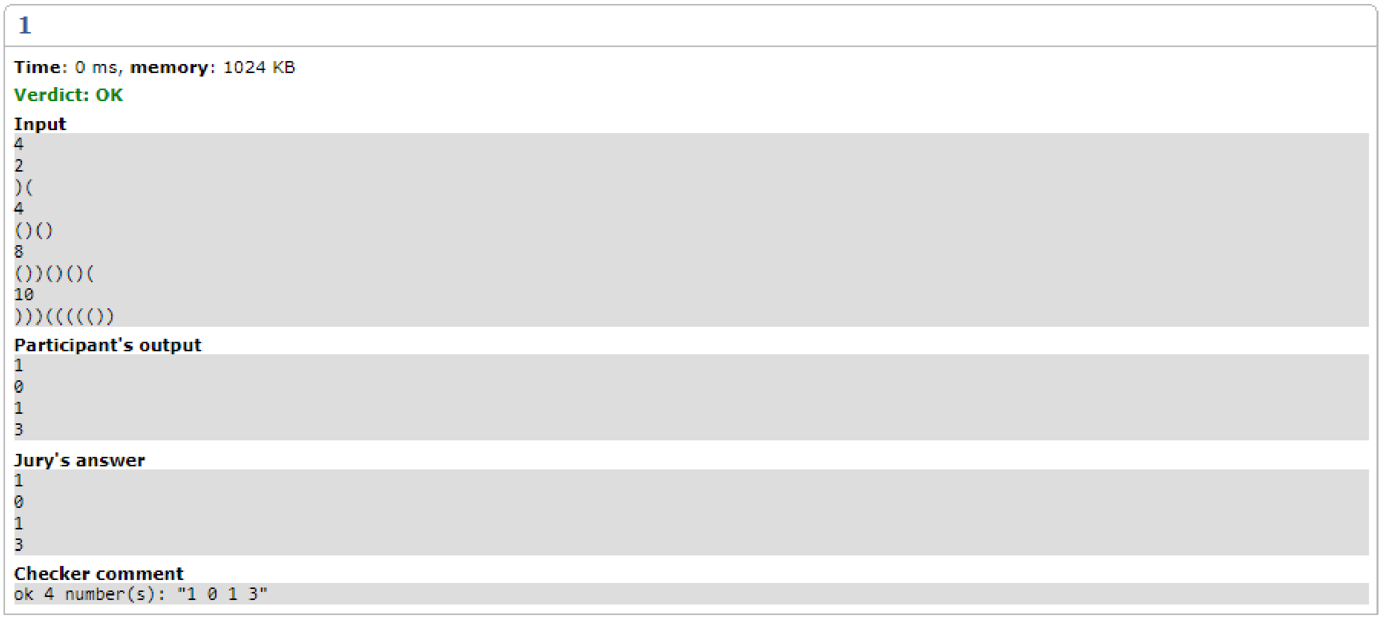
```
#include <iostream>

int main(){

    std::ios_base::sync_with_stdio(false);
    long t; std::cin >> t;
    while(t--){
        long n; std::cin >> n;
        std::string s; std::cin >> s;
        long cur(0), mn(0);
        for(long p = 0; p < s.size(); p++){
            cur += (s[p] == '(') - (s[p] == ')');
            mn = mn < cur ? mn : cur;
        }

        std::cout << (-mn) << std::endl;
    }

    return 0;
}
```

General knowlede used:

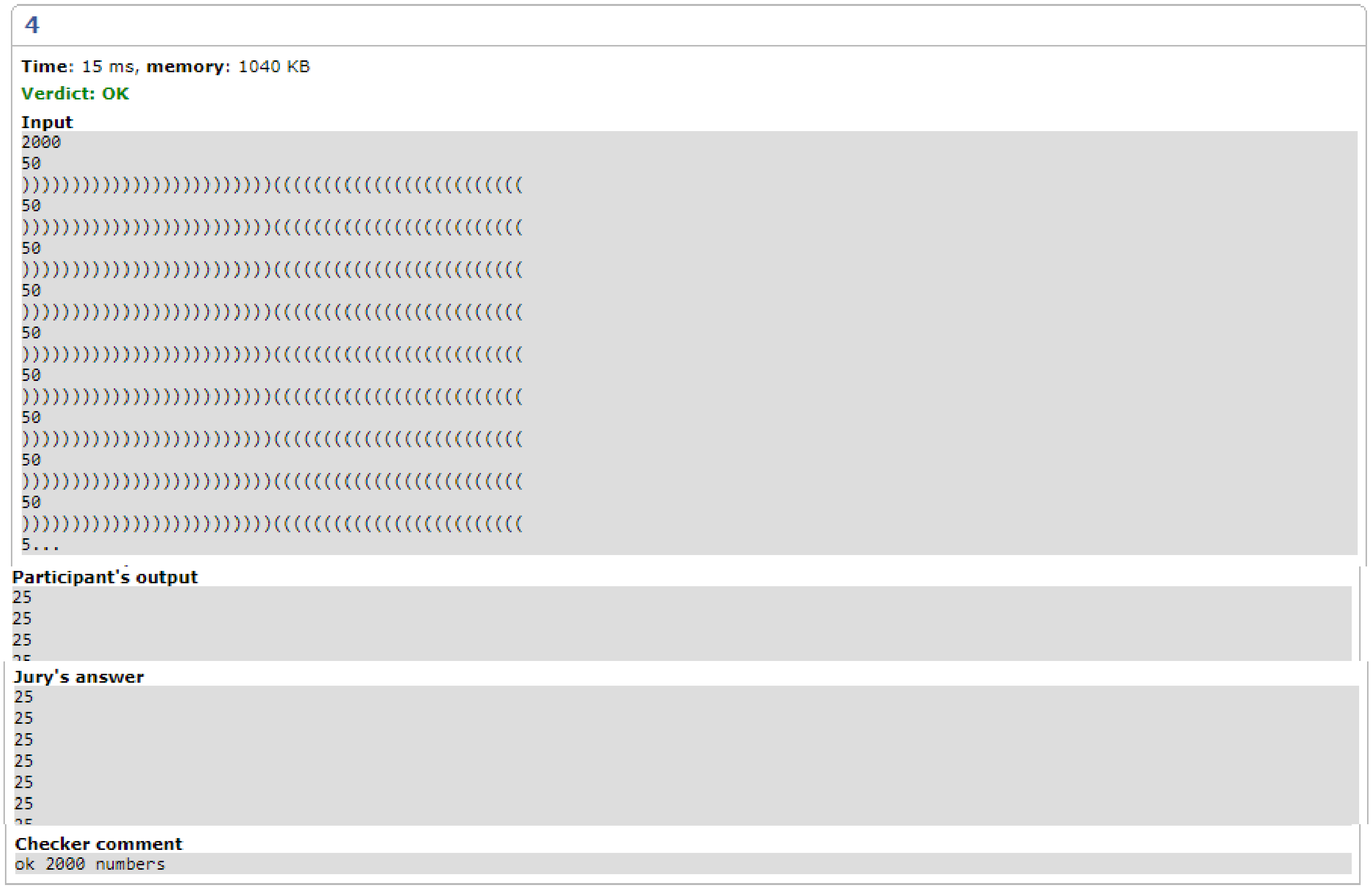
- `#include <iostream>` -iostream is the **header file** which contains all the functions of program like cout, cin etc. and #include tells the preprocessor to include these header file in the program.

1. We define a long type variable `-long t;`, and read it fron console - `std::cin >> t;`, with a” while „ loop we decrement the value of the variable, and in each loop read a new one whe create: `long n; std::cin >> n;` . for using it with 2 new varialbles - `long cur(0), mn(0);`, wich used in the next for loop - `for(long p = 0; p < s.size(); p++){` , to check the paranteses stage and to count the left open onesm wich we print after the for loop ends - `std::cout << (-mn) << std::endl;`

Tests:



```
1

Time: 0 ms, memory: 1024 KB
Verdict: OK
Input
4
2
)(
4
()()
8
())()()(
10
)))(((())
Participant's output
1
0
1
3
Jury's answer
1
0
1
3
Checker comment
ok 4 number(s): "1 0 1 3"
```



```
4

Time: 15 ms, memory: 1040 KB
Verdict: OK
Input
2000
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
50
)))))))))))))))))))))))))(((((((((((((((((((((((((
5...
Participant's output
25
25
25
25
Jury's answer
25
25
25
25
25
25
25
Checker comment
ok 2000 numbers
```

Conclusion:

It’s very interesting to see what you are creating by writing code. It’s like art based on code. Skills were developed to compile, run and test a simple program in the C++ programming language.

As a result of the elaboration of the given paper, the basis was applied for the practical application of the theoretical knowledge.

The structures/concepts/algorithms used in this problem, after writing, compiling the program severaltimes and sending to Codeforces server.

Thus one can judge about the wide possibilities offered by the Java language regarding data manipulation. In this practical work I realized the knowledge accumulated during the theoretical and practical classes, I consolidated the material and in some places I learned new things. It allowed the assessment of knowledge in writing style both for the grade and personally. It allowed us to correct mistakes and possible future misunderstandings. Under the guidance of the teacher, we conducted the first individual study on this subject, this facilitating the adaptation to the knowledge of the use of theoretical material.

The verification of the results confirms that the elaborated program works correctly.

Linear algorithms can be used to calculate mathematical expressions.

Where drawn conclusions about comments, characters, strings, names (identifiers) in C/C++. As in the end I can say that the study had a positive impact on my personal education.

Bibliography:

1) https://en.cppreference.com/w/cpp/container/vector;
2) https://www.geeksforgeeks.org/c-plus-plus/;
3) https://en.cppreference.com/w/cpp/language/goto;
4) https://www.geeksforgeeks.org/vector-in-cpp-stl/;
5) https://www.geeksforgeeks.org/loops-in-c-and-cpp/;
6) https://www.geeksforgeeks.org/selection-sort/;
7) https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/.