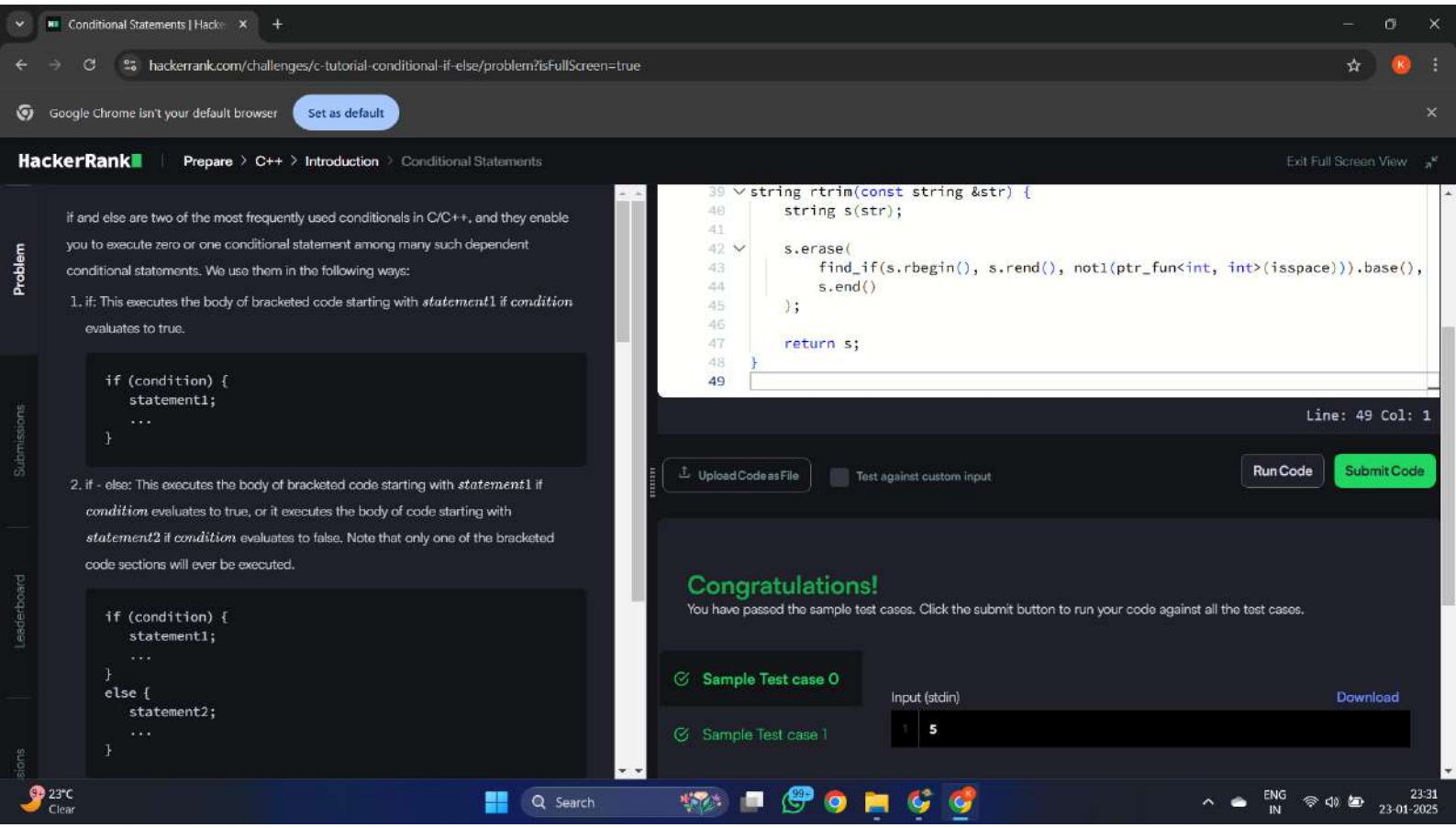
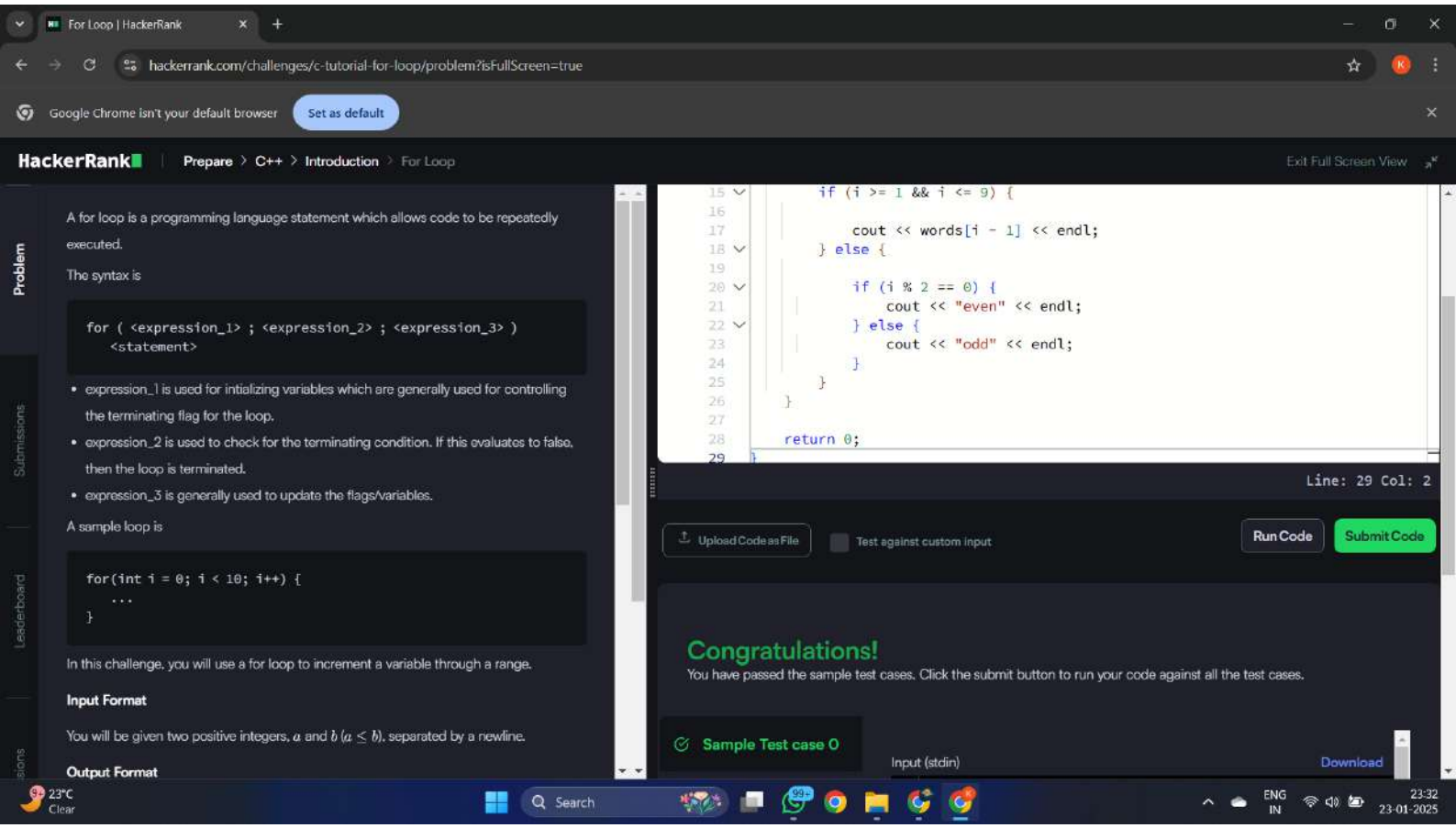
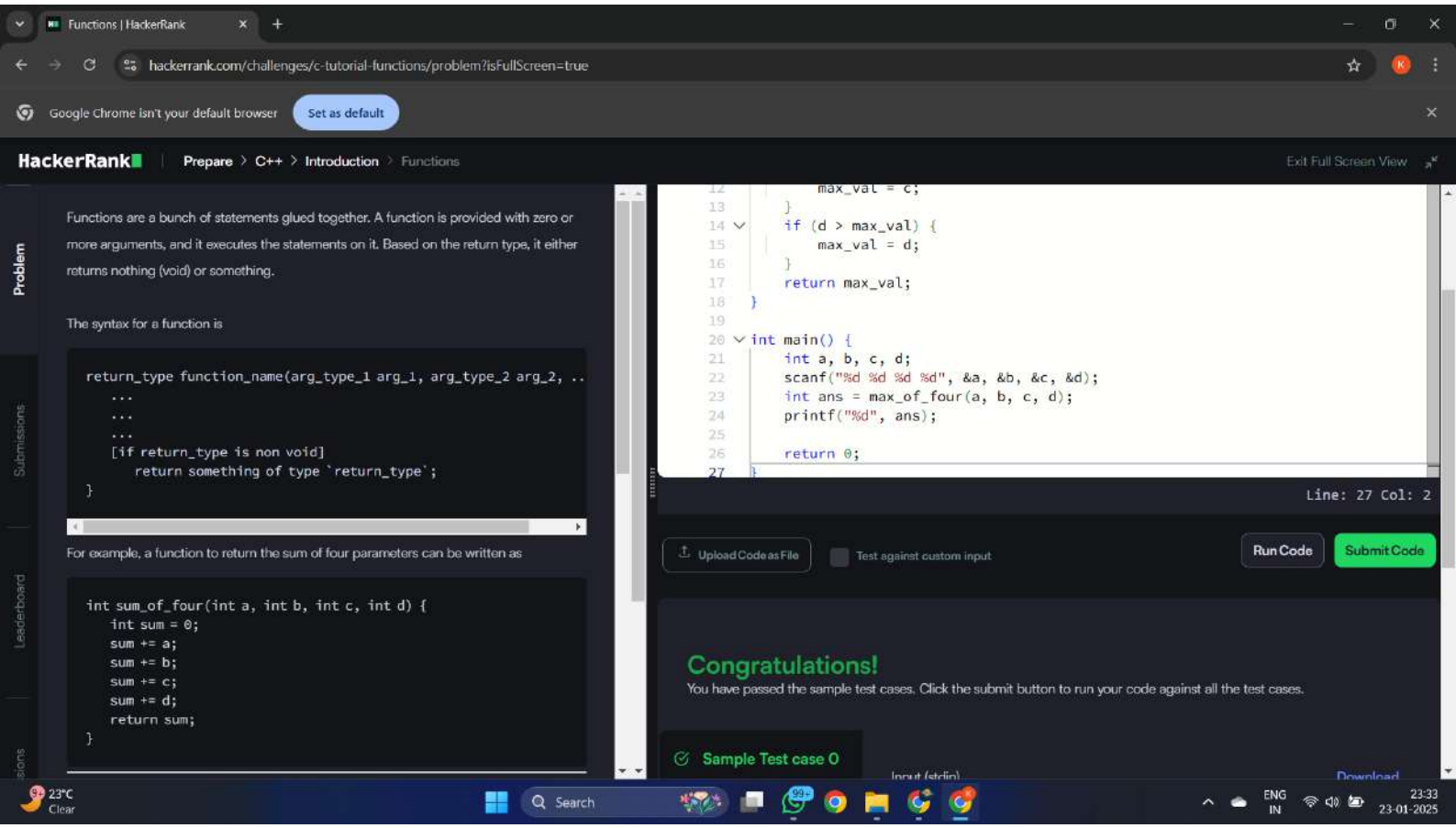
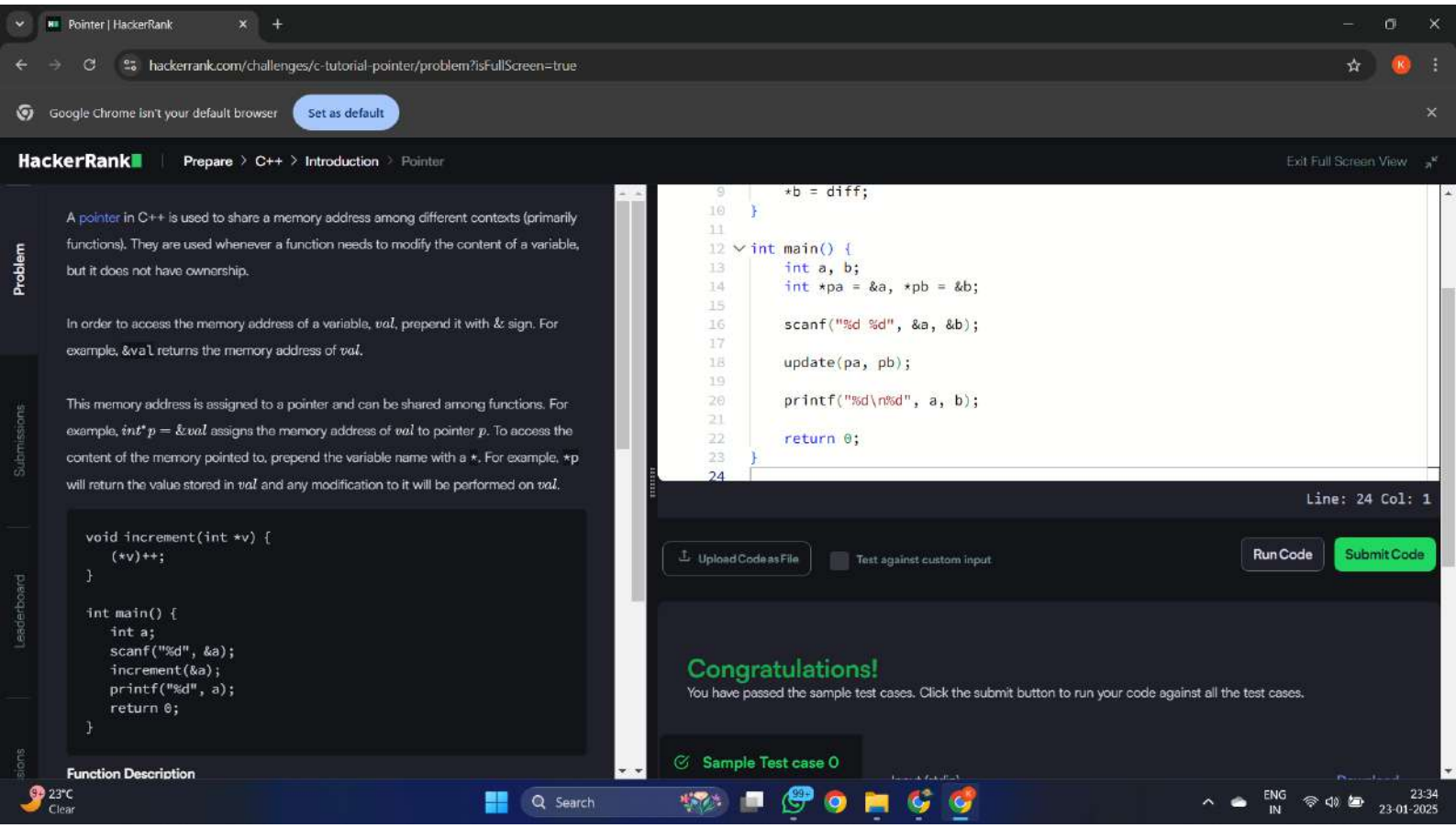
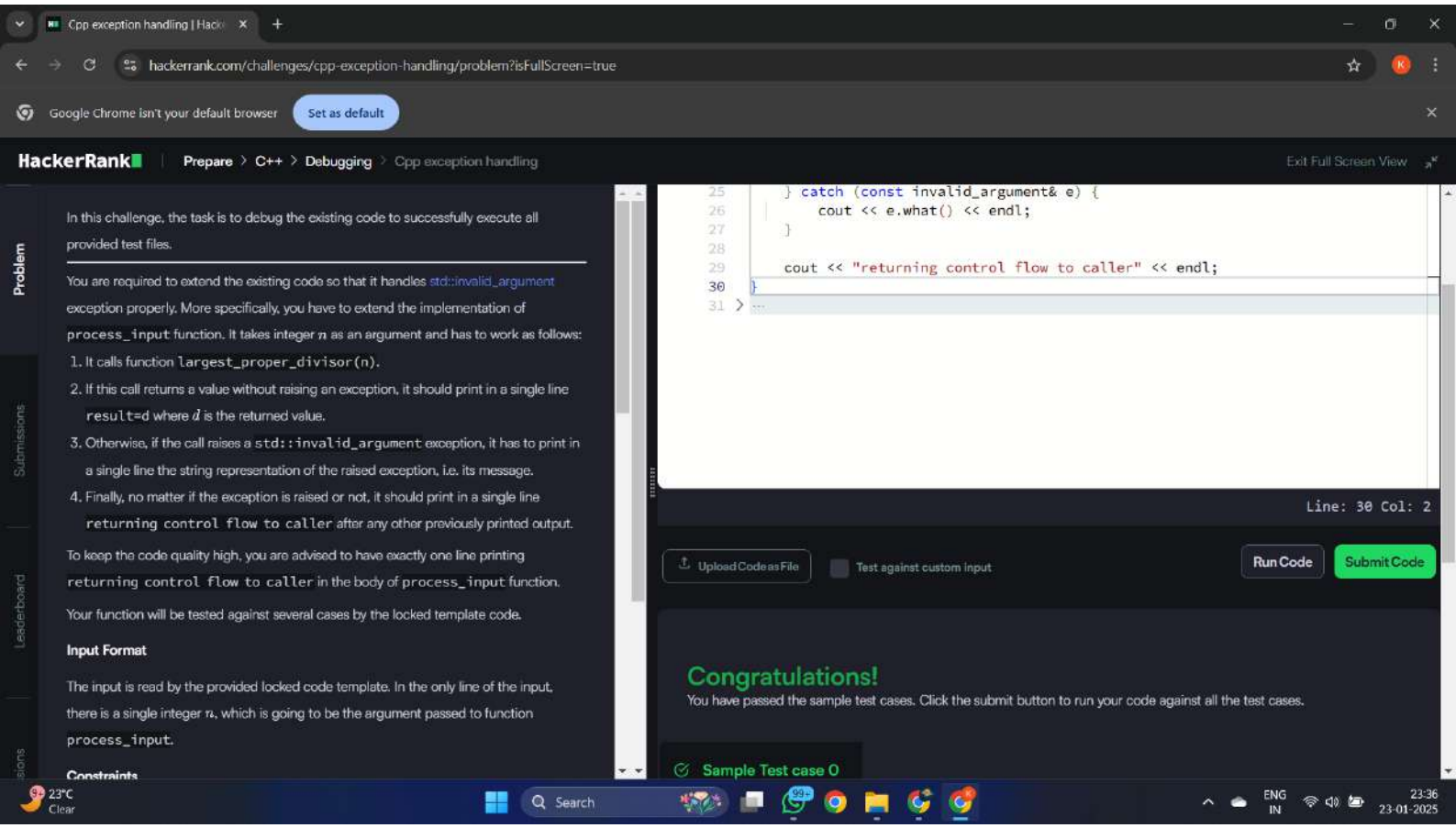
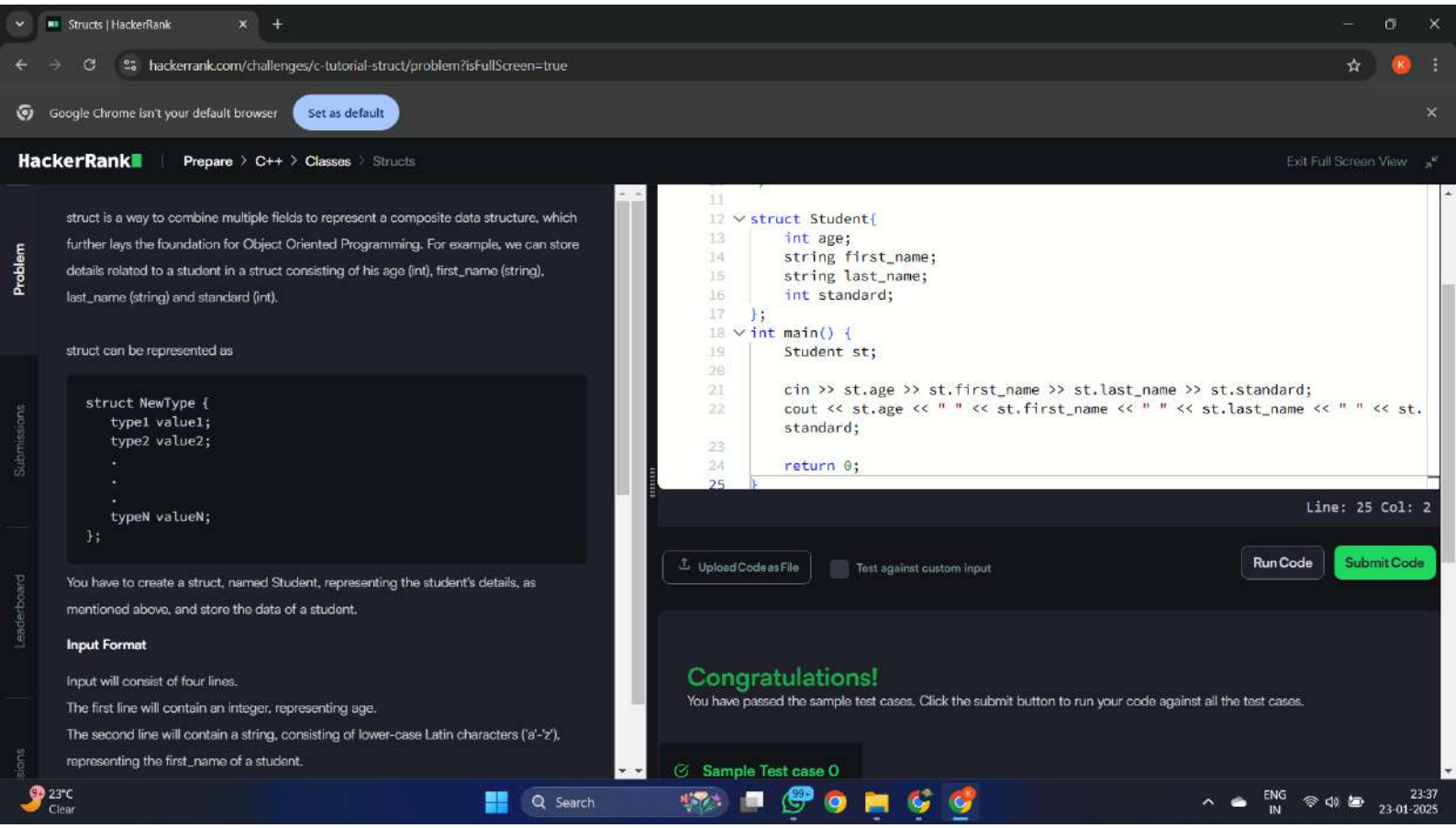

Conditional Statements | Hacke
hackerrank.com/challenges/c-tutorial-conditional-if-else/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Introduction > Conditional Statements
Exit Full Screen View
Problem
Submissions
Leaderboard
if and else are two of the most frequently used conditionals in C/C++, and they enable you to execute zero or one conditional statement among many such dependent conditional statements. We use them in the following ways:
1. if: This executes the body of bracketed code starting with statement1 if condition evaluates to true.
if (condition) {
statement1;
...
}
2. if - else: This executes the body of bracketed code starting with statement1 if condition evaluates to true, or it executes the body of code starting with statement2 if condition evaluates to false. Note that only one of the bracketed code sections will ever be executed.
if (condition) {
statement1;
...
}
else {
statement2;
...
}
string rtrim(const string &str) {
string s(str);
s.erase(
find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
s.end()
);
return s;
}
Line: 49 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Sample Test case 1
Input (stdin)
Download
5
23°C
Clear
Search
ENG
IN
23:31
23-01-2025

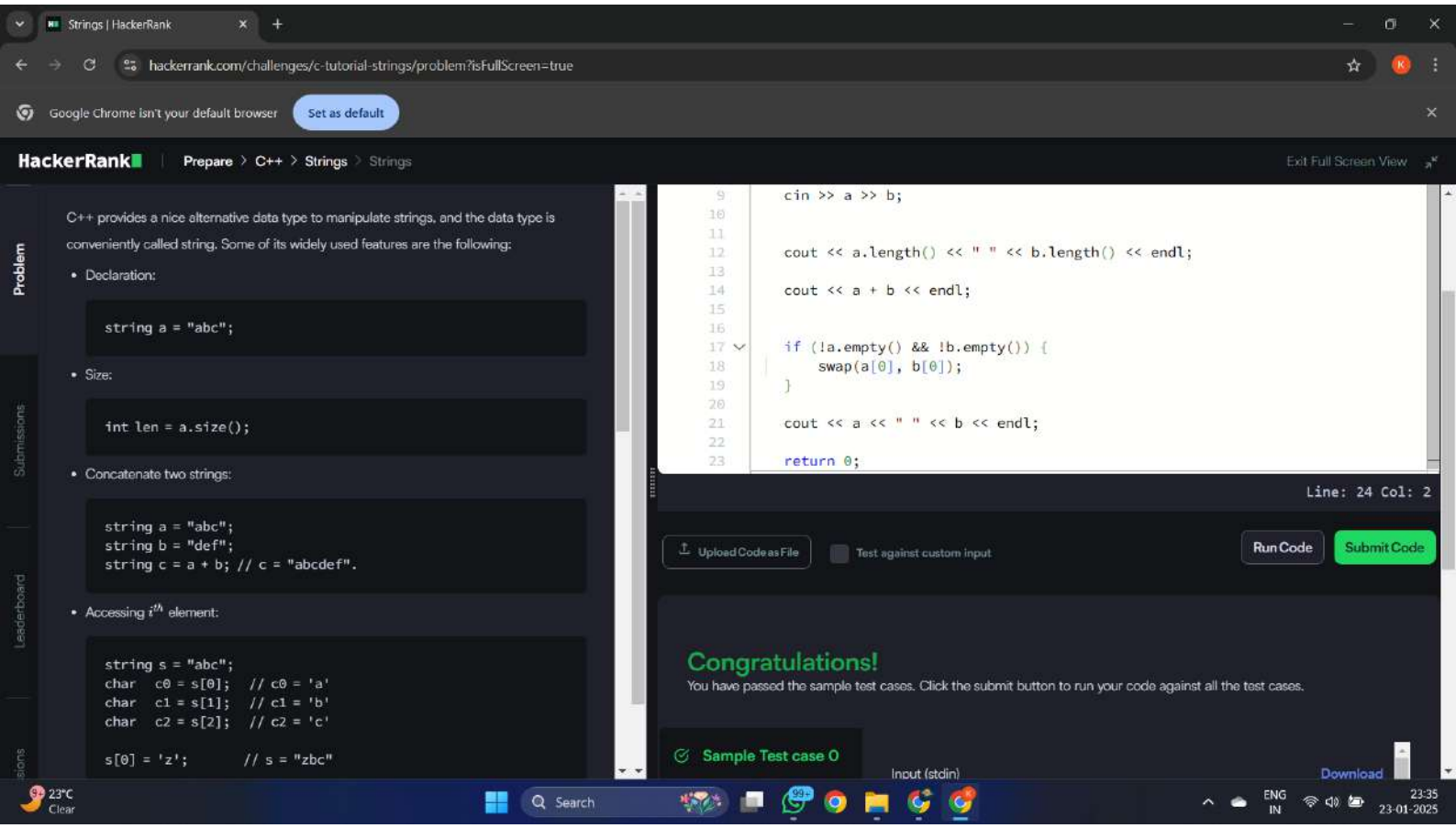

For Loop | HackerRank
hackerrank.com/challenges/c-tutorial-for-loop/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Introduction > For Loop
Exit Full Screen View
Problem
Submissions
Leaderboard
A for loop is a programming language statement which allows code to be repeatedly executed.
The syntax is
for (<expression_1> ; <expression_2> ; <expression_3>)
<statement>
expression_1 is used for intializing variables which are generally used for controlling the terminating flag for the loop.
expression_2 is used to check for the terminating condition. If this evaluates to false, then the loop is terminated.
expression_3 is generally used to update the flags/variables.
A sample loop is
for(int i = 0; i < 10; i++) {
...
}
In this challenge, you will use a for loop to increment a variable through a range.
Input Format
You will be given two positive integers, a and b (a ≤ b), separated by a newline.
Output Format
if (i >= 1 && i <= 9) {
cout << words[i - 1] << endl;
} else {
if (i % 2 == 0) {
cout << "even" << endl;
} else {
cout << "odd" << endl;
}
}
return 0;
}
Line: 29 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
23°C
Clear
Search
ENG
IN
23:32
23-01-2025


Functions | HackerRank
hackerrank.com/challenges/c-tutorial-functions/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Introduction > Functions
Exit Full Screen View
Problem
Submissions
Leaderboard
Functions are a bunch of statements glued together. A function is provided with zero or more arguments, and it executes the statements on it. Based on the return type, it either returns nothing (void) or something.
The syntax for a function is
return_type function_name(arg_type_1 arg_1, arg_type_2 arg_2, ..
...
...
...
[if return_type is non void]
return something of type `return_type`;
}
For example, a function to return the sum of four parameters can be written as
int sum_of_four(int a, int b, int c, int d) {
int sum = 0;
sum += a;
sum += b;
sum += c;
sum += d;
return sum;
}
max_val = c;
}
if (d > max_val) {
max_val = d;
}
return max_val;
}
int main() {
int a, b, c, d;
scanf("%d %d %d %d", &a, &b, &c, &d);
int ans = max_of_four(a, b, c, d);
printf("%d", ans);
return 0;
}
Line: 27 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
23°C
Clear
Search
ENG
IN
23:33
23-01-2025


Pointer | HackerRank
hackerrank.com/challenges/c-tutorial-pointer/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Introduction > Pointer
Exit Full Screen View
Problem
Submissions
Leaderboard
A pointer in C++ is used to share a memory address among different contexts (primarily functions). They are used whenever a function needs to modify the content of a variable, but it does not have ownership.
In order to access the memory address of a variable, val, prepend it with & sign. For example, &val returns the memory address of val.
This memory address is assigned to a pointer and can be shared among functions. For example, int* p = &val assigns the memory address of val to pointer p. To access the content of the memory pointed to, prepend the variable name with a *. For example, *p will return the value stored in val and any modification to it will be performed on val.
void increment(int *v) {
(*v)++;
}
int main() {
int a;
scanf("%d", &a);
increment(&a);
printf("%d", a);
return 0;
}
Function Description
*b = diff;
}
int main() {
int a, b;
int *pa = &a, *pb = &b;
scanf("%d %d", &a, &b);
update(pa, pb);
printf("%d\n%d", a, b);
return 0;
}
Line: 24 Col: 1
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
23°C
Clear
Search
ENG
IN
23:34
23-01-2025


Cpp exception handling | Hacke
hackerrank.com/challenges/cpp-exception-handling/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Debugging > Cpp exception handling
Exit Full Screen View
Problem
Submissions
Leaderboard
In this challenge, the task is to debug the existing code to successfully execute all provided test files.
You are required to extend the existing code so that it handles std::invalid_argument exception properly. More specifically, you have to extend the implementation of process_input function. It takes integer n as an argument and has to work as follows:
1. It calls function largest_proper_divisor(n).
2. If this call returns a value without raising an exception, it should print in a single line result=d where d is the returned value.
3. Otherwise, if the call raises a std::invalid_argument exception, it has to print in a single line the string representation of the raised exception, i.e. its message.
4. Finally, no matter if the exception is raised or not, it should print in a single line returning control flow to caller after any other previously printed output.
To keep the code quality high, you are advised to have exactly one line printing returning control flow to caller in the body of process_input function.
Your function will be tested against several cases by the locked template code.
Input Format
The input is read by the provided locked code template. In the only line of the input, there is a single integer n, which is going to be the argument passed to function process_input.
Constraints
} catch (const invalid_argument& e) {
cout << e.what() << endl;
}
cout << "returning control flow to caller" << endl;
}
Line: 30 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
23°C
Clear
Search
ENG
IN
23:36
23-01-2025


Structs | HackerRank
hackerrank.com/challenges/c-tutorial-struct/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Classes > Structs
Exit Full Screen View
Problem
Submissions
Leaderboard
struct is a way to combine multiple fields to represent a composite data structure, which further lays the foundation for Object Oriented Programming. For example, we can store details related to a student in a struct consisting of his age (int), first_name (string), last_name (string) and standard (int).
struct can be represented as
struct NewType {
typeI valueI;
type2 value2;
.
.
.
typeN valueN;
};
You have to create a struct, named Student, representing the student's details, as mentioned above, and store the data of a student.
Input Format
Input will consist of four lines.
The first line will contain an integer, representing age.
The second line will contain a string, consisting of lower-case Latin characters ('a'-'z'), representing the first_name of a student.
struct Student{
int age;
string first_name;
string last_name;
int standard;
};
int main() {
Student st;
cin >> st.age >> st.first_name >> st.last_name >> st.standard;
cout << st.age << " " << st.first_name << " " << st.last_name << " " << st.
standard;
return 0;
}
Line: 25 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
23°C
Clear
Search
ENG
IN
23:37
23-01-2025


Strings | HackerRank
hackerrank.com/challenges/c-tutorial-strings/problem?isFullScreen=true
Google Chrome isn't your default browser
Set as default
HackerRank
Prepare > C++ > Strings > Strings
Exit Full Screen View
Problem
Submissions
Leaderboard
C++ provides a nice alternative data type to manipulate strings, and the data type is conveniently called string. Some of its widely used features are the following:
Declaration:
string a = "abc";
Size:
int len = a.size();
Concatenate two strings:
string a = "abc";
string b = "def";
string c = a + b; // c = "abcdef".
Accessing i^{th} element:
string s = "abc";
char c0 = s[0]; // c0 = 'a'
char c1 = s[1]; // c1 = 'b'
char c2 = s[2]; // c2 = 'c'
s[0] = 'z'; // s = "zbc"
cin >> a >> b;
cout << a.length() << " " << b.length() << endl;
cout << a + b << endl;
if (!a.empty() && !b.empty()) {
swap(a[0], b[0]);
}
cout << a << " " << b << endl;
return 0;
Line: 24 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
Sample Test case 0
Input (stdin)
Download
23°C
Clear
Search
ENG
IN
23:35
23-01-2025