Code Clash League 2020-2021

REGISTER

www.codeclash.org

CONTACT

teams@codeclash.org

What is Code Clash?

Teams of up to four individuals from high-school and college divisions compete in a challenging, time-limited tournament to put their computer science, problem solving, and teamwork skills to the test. Competitors will be tasked with a series of algorithmic challenges of varying difficulty, while working under the pressure of a strict time limit to produce their solutions in Java, Python, C++, or C.

How does Code Clash League work?

In Code Clash League, there will be multiple remote events played throughout a season, with season-long rankings being tracked for each competing team. Multiple distinct event types will influence competitor rankings, which will eventually determine the qualifications for the end-of-season championship.

Code Clash Sprints

Solo competitors will be given one fast-paced challenge at a time and will have between 5-15 minutes to finish each challenge before moving on.

Code Clash Marathons

Teams of up to four will have between 2-3 hours to solve 10 challenges.

OFFICIAL MANUAL

Languages

The contest supports Java 8, Python 2, Python 3, C++, and C. Competitors may use any combination of these languages to solve the challenges.

Reading and writing

Each challenge will require the program to process an input and return an output. Inputs must be read from "standard in" and outputs must be written to "standard out" (also known as the console). Each problem statement will detail the specific inputs and outputs necessary for each challenge.

Submitting solutions

Competitors will use an online submission system to submit their work to be judged. Competitors will press the green submit button to upload their uncompiled source code to this submission system. The source code file name must only contain alphanumeric characters, underscores, and hyphens. Once the submission is confirmed, the submission list page will be shown with the results.

Viewing the scoreboard

The top of your team page on the submission system shows your team's row in the scoreboard: your position and which challenges you attempted and solved. Via the menu you can view the public scoreboard page with the scores of all teams.

Clarifications

All communication with the judges is to be done through clarifications. These can be found in the right column on your team page on the submission system. Clarification reply broadcasts from the judges are displayed there. There is also a button to submit a new clarification request to the judges; you can ask a general question or one specific to a challenge. This clarification request is only readable for the judges. The judges will send a reply to this clarification to all teams. Teams may be assisted with technical issues unrelated to challenges in person without the need of the clarification system.

Viewing the results of submissions

The left column of your team web page on the submission system shows an overview of your submissions. It contains all relevant information: submission time, programming language, challenge, and status.

CORRECT	The submission passed all tests: you solved this challenge!
COMPILER-ERROR	There was an error when compiling your program. On the submission details page, you can inspect the exact error. Note that when compilation takes more than 30 seconds, it is aborted and this counts as a compilation error.
TIMELIMIT	Your program took longer than the maximum allowed time for this challenge. Therefore, it has been aborted. This might indicate that your program hangs in a loop or that your solution is not efficient enough.
RUN-ERROR	There was an error during the execution of your program. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g. by indexing arrays out of bounds), trying to use more memory than the limit, etc. Also check that your program exits with exit code 0!
NO-OUTPUT	Your program did not generate any output. Check that you write to standard out.
OUTPUT-LIMIT	Your program generated more output than the allowed limit. The output was truncated and considered incorrect.
WRONG-ANSWER	The output of your program was incorrect. This can happen simply because your solution is not correct but remember that your out-put must comply exactly with the specifications of the judges.
TOO-LATE	Bummer, you submitted after the contest ended! Your submission is stored but will not be processed anymore.

Judging submissions

The contest control system uses a fully automated judging system.
Source code submissions will automatically be run through a series of test cases that will determine if the solution is correct. These test cases differ from the publicly visible test cases to prevent hard coding.
After source code is submitted, it enters a queue. The program will be compiled, executed, and tested on one of the auto judge computers running Linux. If the program takes too long to compile, the compilation will be aborted, and the submission will result in COMPILER-ERROR.
Programs will be restricted in CPU time and memory utilization. Programs will be run in a virtual sandbox, where the program will have access to a single CPU core. After execution of the program, the exit status will be checked. Any non-zero exit code will be considered a runtime error, and your submission will result in RUN-ERROR.
The program output will be compared to the test case output. When comparing program output, it must match the format detailed in the problem statement exactly, except for extra whitespace. The auto judge is very strict.

Scoring and individual event rankings

Throughout any individual event, competitors will be ranked first based on how many points they scored, which is determined based on how many problems were completed. The competitor that has completed the most problems will always be ranked first. Ties are broken based on the time score; competitors with lower time scores are ranked higher. The number of minutes since the contest started is added to your time score each time you make a correct submission. As a result, competitors that solve and submit challenges faster will be ranked higher than those who solve the same number of challenges but slower. In addition, a 15-minute penalty for marathons and a 2-minute penalty for sprints is added to the time score for each incorrect submitted attempt if that challenge ends up being solved (no penalty is given if that problem is never correctly solved).

RANK	TEAM	SCORE		A	B	C	D	E
6	Ukkonen Fan Club University of Helsinki	3	112	1 try	22 1 try		42 2 tries	1 try

Challenges

Each contest will be composed of a series of challenges. Each challenge will have equal point value, although they may vary in difficulty. Each challenge will have a problem statement, where the challenge is explained in detail. The format of inputs and expected outputs will be detailed in each problem statement. At least one sample test case will be included with each problem statement. Competitors must work to develop a single solution to each challenge that will produce the desired result.

Sprint Events

In a sprint event, one challenge will be presented at a time. As soon as the time expires for each specific challenge, no more submissions will be accepted for that challenge and everyone will move on. Each challenge will be presented for between 5 to 15 minutes, depending on the difficulty. A few minutes of rest time will be provided between each sprint challenge. Each sprint event is expected to last about 90 minutes and feature several challenges.

Sprint events are solo events; therefore, no collaboration is allowed inside or outside of one's team, including with teachers or instructors.

The top two performing individuals from each team will determine the overall team score in the sprint events.

Marathon Events

In a marathon event, ten challenges will be released at the start of the event. Competitors will have 150 minutes to solve and submit as many challenges as they can. Challenges will vary in difficulty, but competitors can skip around through the challenges and choose which ones to start first.

Marathon events are team events, but no collaboration is allowed outside of one's team, including with teachers or instructors.

Software, IDEs, and Resources

Competitors may compete from their local school or from their homes and may use any computers with any appropriate software during the contest. Constant internet access is required for the event. Any printed resources, notes, textbooks, or online resources are allowed and may be used.

Each competitor is responsible for ensuring that they have access to a computer with the necessary software installed for writing source code. Any modern browser will be capable of accessing the Code Clash web portal.

Divisions

All competitors will be classified into one of two divisions by their age as of January 1, 2021. Competitors will compete directly against those in the same age division as them. Upon registration, competitors in the 13 to 18 Division will be asked to verify their age by providing a teacher endorsement, verifying with their school email address, or providing some other document to prove age.

- 13 to 18 Division (High School Division)
- 19 and Up Division (College and Beyond Division)

Up to four competitors may compete on a team together. Teams of less than four competitors are acceptable. In the event that there are competitors on a team across multiple age divisions, the team will compete in the highest age division.

League Rankings

// TODO: Design Ranking Points system for the league leaderboard.

For sprint events, competitors compete solo and will each receive their own individual rankings for the event, calculated from their number of challenges solved and time score. Using an algorithm based on ranking and number of competitors at the particular event, each individual will score a certain number of Ranking Points (RP). The top two performing individuals in RP from each team will have their Ranking Points added together to form the team's RP score for that sprint event.

For marathon events, teams compete together and will have one combined ranking for the event, calculated from their number of challenges solved and time score. Using an algorithm based on ranking and number of teams at the particular event, each team will score a certain number of Ranking Points.

Teams are allowed to participate in as many events as they wish. In the overall league rankings, teams are ranked by their top two sprint event performances and their top marathon event performance. Ranking Points from these three events are added up to calculate the overall league rankings for the entire season.

LANGUAGE REFERENCE

Below is an example challenge that demonstrates standard in/out usage.

The first line of input contains the number of testcases. Each testcase consists of a line containing a name (a single word) of at most 99 characters. For each testcase, output the string "Hello <name>!" on a separate line.

Sample test case:

Input	Output
3 world Jan SantaClaus	Hello world! Hello Jan! Hello SantaClaus!

Solution code in all supported languages is found on the next few pages. The demonstrated format for reading from standard in and writing to standard out will be the default procedure for all the challenges at Code Clash. All problems will follow this basic format.

A solution for this challenge in Java:

```java
import java.util.*;

class Main {
        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);
                int nTests = scanner.nextInt();

                for (int i = 0; i < nTests; i++) {
                        String name = scanner.next();
                        System.out.println("Hello " + name + "!");
                }
        }
}
```

A solution for this challenge in Python:

```python
n = int(input())
for i in range(n):
    name = input()
    print('Hello ' + name + '!')
```

A solution for this challenge in C++:

```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
        int ntests;
        string name;

        cin >> ntests;
        for (int i = 0; i < ntests; i++) {
                cin >> name;
                cout << "Hello " << name << "!" << endl;
        }
}
```

A solution for this challenge in C:

```c
#include <stdio.h>

int main() {
        int i, ntests;
        char name[100];

        scanf("%d\n", &ntests);

        for (i = 0; i < ntests; i++) {
                scanf("%s\n", name);
                printf("Hello %s!\n", name);
        }
}
```

OFFICIAL RULES

1. Eligibility

All Code Clash competitors must be at least 13 years of age. A Code Clash team must not exceed four competitors, although any number of mentors or instructors can be associated with a given team. Mentors can be associated with more than one team and can be considered a competitor themselves on at most one team.

All Code Clash teams must be classified within one of the two divisions: 13 to 18 Division or 19 and Up Division. The team is classified based on the age of its oldest member. For teams wishing to compete in the High School Division, all team members must provide some form of age verification prior to their first event; teams that do not provide age verification will only be able to compete in the 23 and Up division.

High School Division teams are highly encouraged to verify their enrollment in a high school by retrieving an endorsement from a teacher at their school. Alternatively, team members can use their school email addresses or provide some other document that verifies their age or high school enrollment. These verification documents should be submitted via email to teams@codeclash.org after registration.

Teams are allowed to register at any point in the season, but High School Division teams should register and provide age verification at least 48 hours prior to their first event if they want to guarantee they will be able to compete. Late registrations may be forced to compete in the 19 and Up Division if there is insufficient time for verification.

Teams must register at the Code Clash website (www.codeclash.org) or contact our email address if further help is needed (teams@codeclash.org).

By registering, all competitors agree to the entirety of the rules/manual and the terms listed below:

I hereby grant permission to Code Clash and its organizers to use and publish my name, my school/organization, and my scoring information associated with the event. I hereby release any and all claims against any person or organization utilizing this material for purposes deemed appropriate by Code Clash.

Code Clash officials have the right to override any call or rule in this manual. All decisions by Code Clash officials are final.

This rules manual is subject to change at any time without notice.

2. Conduct Rules

Mentors and/or spectators may accompany the competitors in the competition, but they are not to assist or collaborate with the competitors for the duration of the contest. Competitors may not communicate with anyone outside their team in person, through the internet, or through any other means. An attempt to collaborate with someone outside of one's team may result in disqualification. Competitors are expected to be respectful to other competitors and officials. Misbehavior will be investigated and may result in disqualification.

3. Necessary Materials

Competitors are responsible for providing their own computational device with a stable internet connection. Competitors are responsible for installing any necessary or desired software such as IDEs, compilers, or browsers. Any IDE, compiler, or browser is allowed for use during the competition.

Any printed resources such as notes, cheat-sheets, or textbooks are allowed. Additional computational devices such as calculators are also allowed.

Internet resources such as official documentation, internet forums, or tutorial websites are allowed. Note that posting a specific question on an internet forum would be a violation of our collaboration rules per Section 2.

4. Computer Rules

Competitors should always have their web browser opened to the Code Clash submission portal for the full duration of the event. When submitting solutions to challenges, competitors must upload their uncompiled source code to this submission portal. The submission portal accepts Java, C++, C, Python 2, and Python 3 source code.

Source code must be a single file not exceeding 256 kilobytes. Compilation of your program may take no longer than 30 seconds. During execution of your program, there are 524288 kilobytes of memory available. Programs are limited to no more than 5 seconds of CPU time and 1 thread.

All code will be compiled and tested per the details in the official manual.

5. General Contest Rules

The main problem statements will be released publicly at either the start of the contest (marathon events) or at the start of the challenge period (sprint events), and the competitors may immediately begin their work on a computer. Competitors will code their solutions to each of the challenges and submit them on the submission portal.

If a competitor has a clarification question related to one of the challenges, they must submit their question through the clarification system on the submission portal. Contest officials may not be able to answer questions related to the challenges, although they may assist competitors to use the submission system or troubleshoot other issues.

When a solution is submitted to a challenge, it will be automatically judged, and the submitter will receive feedback on whether their solution passed all test cases. If the solution fails, the submitter will be notified, and they will be able to modify their code and submit an infinite number of attempts. If all test cases are passed, the submitter will be credited with completing that challenge and the total number of minutes since the start of the competition will be added to their time score.

Teams and competitors will be ranked first based on the number of challenges they have completed, and second based on their time score. The team or competitor with the most completed challenges will be ranked first. In the event of a tie, the team or competitor with the lowest time score will be ranked higher (think golf scores).

Penalty time will be added to the time score for each submitted failed attempt to a challenge, including compile errors, runtime errors, and wrong answers. 10 minutes of penalty time is added for each failed attempt in marathon events, and 1 minute of penalty time is added for each failed attempt in sprint events.

A leaderboard will be shown to both competitors and observers for the first half of marathon events. For the second half of marathon events, the leaderboard will be frozen. The final leaderboard and standings will be revealed to both competitors and mentors at the conclusion of the event.

A leaderboard for each individual challenge will be shown during sprint events, and the combined leaderboard will be revealed at the conclusion of the event.