



Эслатма

Если не удастся отправить решение через CMS необходимо предпринять следующие действия:

- Скопируйте ваши отправляемые файлы в директорию с названием submit_<task name> на рабочем столе (директория должна быть создана) **до конца соревнования**.
 - В задачах с частичным оцениванием выходных файлов, Ваши файлы должны быть названы как 01.out, 02.out, ..., 10.out.
 - Для остальных задач, должен быть **ровно один** файл в этой папке, содержащий Ваш исходный код.
- Попросите Вашего лидера делегации отправить апелляцию.

Агар удаётся эчимни юборишга CMS предпринять даркор янаги амалнинг:

- скопируйте сизнинглар жўнатлар директориясига файллар названиемдан submit_ ишчи столдасига(директория талаб этилган бўлиш создана) тамомнинг **дои соревнования**.
- топшириқ - частичным оцениваниемдан чиқадиган файлларнинг, Сизнинглар файлларингиз талаб этилган бўлиш названы қандай 01. out, 02. out, ..., 10. out.
- бошқа топшириқ учун, талаб этилган **ровно бўлиш один** файл ушбу папкеига, асрамоқ Сизнинг дастлабки кодингиз.
- Сизнинг лидери попросите вакилларнинг апелляцию юбориш.

Если вы считаете, что Вам необходимо предоставить дополнительное время, выполняйте следующие действия:

- Отправьте вопрос (либо через веб-интерфейс, либо на бумаге) как можно раньше
- Не покидайте своего места и не разговаривайте с другими участниками по окончанию соревнования.

Агар сизлар ҳисаблайсиз, нима Сизларга предоставить даркор қўшимча вақт, ўзини жангга тузайсиз янаги амалнинг:

- савол Юборасиз(ёки веб-интерфейс, ёки қоғозига) қандай
- покидайтема ўзнинг жойи ва бошқа иштирокчилардан гапиришма мусобақанинг окончанию бўйича.

У каждого задания есть приложение, которое можно скачать через веб-интерфейс.

Хар бир вазифада илова бўлади, қайсики скачати мумкин веб-интерфейс.

Для заданий с проверкой выходных файлов:

- Приложение содержит входные данные и примеры тестов. Каждый тест является отдельной подзадачей.
- Вы можете посылать несколько выходных файлов одним zip-архивом. Для этих целей Ваши файлы должны быть названы `?.out`, где `??` является номером теста (т.е., `03.out`). Вы можете заархивировать несколько файлов используя следующую команду: `zip output.zip *.out`
- Вы можете сделать не более 100 посылок для задач с проверкой выходных данных. В каждой посылке разрешается отправить выходные файлы для любого подмножества тестов.

Вазифа учун чиқадиган файлларнинг синашдан:

- Илова асрайди кирши маълумотларлар ва тестов ибратлари. Ҳар бир тест бўлади алоҳида подзадачей.
- Сизлар йўлла оласиз алланеча чиқадиган файлларнинг бир zip-архивом. ушбу мақсад учун Сизнинглар файлларингиз талаб этилган бўлиш названы `?.out`, қаерда `??` номером бўлади теста(т. е., `03.out`). Сизлар заархивировать оласиз алланеча файлларнинг фойдаланмоқ янаги командани: `zip output.zip *.out`
- Сизлар қила оласиз кўпроқ эмас 100 топшириқ учун посилкаларнинг чиқадиган маълумотларларнинг синашдан. ҳар бир посилкасига юборишга эчилади чиқадиган файллар тестов ҳар бир подмножества учун.

Для других задач:

- Приложение содержит примеры проверяющего модуля, примеры реализаций, примеры тестов и скрипты компиляции.
- Вы можете сделать до 50 посылок по каждой задаче, каждая посылка должна содержать ровно один файл.
- Имя файла для отправки дано в заголовке условия задачи. Он должен реализовать функции, описанные в условии задачи используя предоставленные в примерах реализации сигнатуры.
- Разрешено реализовывать любые другие функции

Бошқа топшириқ учун:

- Илова ибратларни асрайди текширинг модуля, реализаций ибратлари, тестов ибратлари ва компиляциянинг скрипты.
- Сизлар қила оласиз 50 посилкаларнинг ҳар бир топшириқ бўйича, ҳар бир посилка талаб этилган нақини асраш бир файл.

- отправки учун файлнинг Исми берилган шартнинг сарлавҳасига топшириқнинг. У талаб этилган функции реализовать, описанные топшириқнинг шартига фойдаланмоқ предоставленные реализации ибратларига сигнатуры.
- разрешено реализовывать ҳар бирлар бошқалар функции
- Посылкам запрещено считывать из стандартного потока ввода, писать в стандартный поток вывода, или взаимодействовать с любым другим файлом. Тем не менее, они могут писать в стандартный поток ошибок.
- Посылкам **запрещается вызывать `exit()` или `System.exit()`.**
- При тестировании программ с помощью примера проверяющего модуля, входные данные должны соответствовать формату и ограничениям согласно условию задачи. В противном случае результаты непредсказуемы.
- Во входных данных примера проверяющего модуля, любые два последовательных токена в строке разделены единственным пробелом, за исключением случаев когда явно обговорено обратное.
- Мы рекомендуем использовать скрипты из приложения при тестировании кода на локальной машине. В противном случае, особенно в C++, убедитесь что выставлен флаг компиляции `=std=gnu++14`.
- Посилкаларга тақиқланган из считывать стандартного потока киритилишнинг, мирзаш стандартный потокига хулосанинг, ёки бир-бирига ёрдамлашиш ҳар бир бошқа файл билан. тем менее эмас, уларлар мирза оладилар стандартный потокига хатоларнинг.
- Посилкаларга **чақиришни запрещается `exit()` ёки `System.exit()`.**
- программ тестированиида ибратнинг кўмак билан текширнинг модуля, кирши маълумотларлар талаб этилган соответствовать формату ва ограничениям топшириқнинг шартига. ёқимсиз ҳодисадасига натижалар непредсказуемы.
- кирши маълумотларларнинг ибратнинг текширнинг модуля, ҳар бирлар икки изчил токена сатр - разделены ягона пробелом, ҳодисаларнинг мустасно билан қачон явно обговорено тескари.
- Бизлар скрипты фойдаланиш тавсия қиламиз тестировани при илова из коднинг маҳаллий машинеисига. ёқимсиз ҳодисадасига, айниқса C++, убедитесь нима кўрсатилган компиляциянинг флаги `=std=gnu++14`.

Битимнинг

Условия задач задают сигнатуры используя общие названия типов данных `void`, `int`, `int64` и `int[]` (массив).

Топшириқларнинг шартлари сигнатуры задают фойдаланмоқ умумий названия тиканларнинг берларнинг `void`, `int`, `int64` ва `int[]` (массив) .

Проверяющие модули используют следующие типы данных или реализации для

каждого из поддерживаемых языков программирования:

Текширлар модули фойдаланадилар янаги тиканлар маълумотларларнинг ёки реализации ҳар бир из учун қўллаларнинг программирования тили:

Тил	void	int	int64	int[]	массивнинг бўйи a
C++	void	int	long long	std::vector<int>	a.size()
Java	void	int	long	int[]	a.length

Ограничения

Топшириқ	Вақт бўйича ограничение	Эс бўйича ограничение
line	проверка выходных файлов	проверка выходных файлов
vision	1 сек	1024 MB
walk	4 сек	1024 MB