**Регламент**

турнира “Регулярные соревнования по спортивному программированию”

**Раздел 1. Общие положения**

1.1.Определение победителя Турнира “Регулярные соревнования по спортивному программированию” (далее - Турнир);

1.2. Подготовка спортсменов к участию в региональном турнире по спортивному программированию;

1.3. Отработка формата и технологий проведения соревнованиях по спортивному программированию;

1.4. Повышение уровня мастерства спортивных программистов;

**Раздел 2. Место и сроки проведения физкультурного мероприятия**

2.1. Организацию и контроль за проведением Турнира осуществляет Организационный комитет (далее - Оргкомитет), состав которого утверждается ФСП;

2.2. В случае спорных ситуаций, которые не могут быть урегулированы положениями настоящего Регламента, Оргкомитет вправе принять решение, которое будет обязательно для всех участников Турнира;

2.3. Сроки проведения Турнира: 1-5 июня 2022 г. ;

2.4. Место проведения Турнира: онлайн в сети интернет на платформе по адресу https://bacs.cs.istu.ru/

**Раздел 3. Требования к участникам физкультурного мероприятия и условия их допуска**

3.1. К участию в Турнире допускаются только спортсмены достигшие 18 лет и подавшие заявки на

3.2. Турнир имеет личный характер проведения;

**Раздел 4. Правила соревнования**

4.1 Это соревнование проводится по правилам спортивного программирования.

4.2 Продолжительность соревнований составляет 5 часов. Оргкомитет имеет право продлить соревнование в случае возникновения непредвиденных обстоятельств.

4.3 Участникам будет предложено для решения 5 задач. Все задачи сформулированы на русском языке. Для решения задач не требуются какие-либо глубокие знания в конкретной прикладной области или совершенное владение одним из языков программирования.

4.4 Каждая задача предполагает ввод исходных данных из файла INPUT.TXT и вывод результата в выходной файл OUTPUT.TXT. Входной и выходной файлы располагаются в текущей директории.

4.5 В течение соревнования участники решают предложенные задачи. Решением является исходный код на одном из доступных языков программирования, при этом различные задачи могут решаться на разных языках программирования.

4.6 Решения посылаются на проверку в течении соревнования с помощью специального программного обеспечения. После отправки на проверку одного из решений участники могут продолжать работать над другими задачами.

4.7 Решения проверяются прямо во время соревнования. Как только проверка очередного решения завершается, участник тут же уведомляется о её результате. Если решение не принято, участник может попытаться исправить свою программу и послать её на проверку ещё раз.

**Раздел 5. Проверка решений**

5.1 Каждое решение, отосланное на проверку, проверяется путем запуска на секретном наборе тестов, одинаковом для всех участников. Задача считается решённой только в том случае, если решение выдаёт верные ответы на все тесты из набора.

5.2 Тестирование производится автоматически, поэтому программы должны в точности соблюдать форматы входных и выходных файлов, описанные в условии задачи.

5.3 Для каждой задачи будет установлено максимальное время выполнения программы на каждом тесте и максимальное количество используемой программой памяти. Решение не засчитывается, если программа не укладывается в указанные лимиты времени и памяти.

5.4 После каждой проверки решения Оргкомитет отправляет участнику, предоставившему решение, сообщение с результатами тестирования. Участнику сообщается, зачтено решение или нет. Если решение не зачтено, сообщается тип ошибки и номер теста, на котором произошла ошибка.

5.5 В ответ на посланное решение, участник получает один из следующих ответов от проверяющей системы:

* Ошибка компиляции. (Compilation error) Оргкомитету не удалось провести компиляцию вашей программы.
* Превышен лимит времени. (Time limit exceeded) Время работы программы превышает максимально допустимое для данной задачи (указано в условии) либо программа требует ввода данных с клавиатуры.
* Превышен лимит памяти. (Memory limit exceeded) Объём используемой программой памяти превышает максимально допустимый для данной задачи (указано в условии).
* Аварийное завершение программы. (Runtime error) Программа аварийно завершилась, либо код возврата не нулевой. Чтобы код возврата был нулевой, используйте "return 0;" в С и С++ программах и не используйте "halt(n)" в PAS программах, где n отлично от нуля.
* Неверный формат вывода. (Presentation error) Выходной файл программы не был создан в результате её выполнения, либо программа осуществляет вывод данных на экран.
* Неверный ответ. (Wrong answer) Содержимое выходного файла вашей программы не является корректным ответом для предложенного входного файла.
* Зачтено. (Accepted) Задача засчитана. Переходите к решению других задач.

5.6 Номер ошибочного теста, указываемый в результатах проверки, есть номер первого теста, не прошедшего проверку.

5.7 Оргкомитет может прекратить информирование команд о результатах проверки решений за несколько минут до конца соревнования, если возникнет ситуация перегрузки проверяющей системы. Тем не менее, все решения, присланные на проверку до завершения соревнования, будут протестированы Оргкомитетом после окончания соревнования и учтены в финальном протоколе.

5.8 Набор тестов, на которых Оргкомитет проверяет решения, не предоставляется участникам даже после окончания соревнований.

**Раздел 7. Подведение итогов**

7.1 Оргкомитет обладает исключительным правом определения правильности присланных решений, определения победителей и дисквалификации команд. Директор соревнований и Оргкомитет разбирают вопросы, возникшие в результате непредвиденных событий и обстоятельств. Решения Оргкомитета окончательны и обжалованию не подлежат.

7.2 При подведении итогов соревнования используются следующие правила. Выигрывает участник, набравший наибольшее количество баллов. В случае равенства количества полученных баллов, участник с меньшим штрафным временем занимает более высокое место. Временем решения задачи считается время получения Оргкомитетом для тестирования правильного решения задачи. От времени решения задачи высчитываются штрафные баллы, за каждую минуту потраченную на решение, но не более 70% от максимальной стоимости задачи.Отправка неверного решения на проверку Оргкомитету добавляет 50 штрафного баллов за каждую попытку. На задачу, которая не решена, не начисляются штрафные баллы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Задача | Макс. балл | Мин. балл (30%) | Потеря (баллов в минуту) | Балл на конец турнира |
| A | 500 | 150 | 1 балл | 200 |
| B | 1000 | 300 | 2 балла | 400 |
| C | 1500 | 450 | 3 балла | 600 |
| D | 2000 | 600 | 4 балла | 800 |
| E | 2500 | 750 | 5 баллов | 1000 |

Таблица №1 - Распределение баллов за задачи на турнире

**Раздел 8. Формат Турнира**

8.1. Соревновательной платформой Турнира является https://bacs.cs.istu.ru/

8.2. Турнирной платформой осуществляется поддержка следующих языков (для каждого указана строка компиляции и/или запуска):

* [gcc] COLLECT\_GCC=gcc COLLECT\_LTO\_WRAPPER=/usr/lib/gcc/x86\_64-unknown-linux-gnu/4.7.2/lto-wrapper Целевая архитектура: x86\_64-unknown-linux-gnu Параметры конфигурации: /build/src/gcc-4.7.2/configure --prefix=/usr --libdir=/usr/lib --libexecdir=/usr/lib --mandir=/usr/share/man \ --infodir=/usr/share/info --with-bugurl=https://bugs.archlinux.org/ --enable-languages=c,c++,ada,fortran,go,lto,objc,obj-c++ \ --enable-shared --enable-threads=posix --with-system-zlib --enable-\_\_cxa\_atexit --disable-libunwind-exceptions --enable-clocale=gnu \ --disable-libstdcxx-pch --enable-libstdcxx-time --enable-gnu-unique-object --enable-linker-build-id --with-ppl --enable-cloog-backend=isl \ --disable-ppl-version-check --disable-cloog-version-check --enable-lto --enable-gold --enable-ld=default --enable-plugin --with-plugin-ld=ld.gold \ --with-linker-hash-style=gnu --disable-multilib --disable-libssp --disable-build-with-cxx --disable-build-poststage1-with-cxx --enable-checking=release
Модель многопоточности: posix
gcc версия 4.7.2 (GCC)
* [java] extra/openjdk6 6.b24\_1.11.4-1
* [C++] compile=/usr/bin/c++ -O2 -fno-stack-limit -x c++ {src} -o{src}.o
* [C++11] compile=/usr/bin/c++ -O2 -fno-stack-limit -std=c++0x -x c++ {src} -o{src}.o
* [C] compile=/usr/bin/c++ -O2 -fno-stack-limit -x c {src} -o{src}.o
* [FPC (Delphi mode)] /usr/bin/fpc -Cs`echo '64\*2^20-1025' | bc` -Xt -O2 -Mdelphi {src} -o{src}.exe
* [FPC] /usr/bin/fpc -Cs`echo '64\*2^20-1025' | bc` -Xt -O2 -Mfpc {src} -o{src}.exe
* [Java] javac -g:none -classpath %s.dir %s.dir/%s" // note: public class Main {} is required