

Problemas Interactivos y DMOJ

Los problemas de esta semana serán problemas interactivos de ediciones anteriores de la olimpiada internacional. Hemos decidido hacer un entrenamiento de este tipo porque en esta edición de la OIE pueden caer problemas interactivos y estaría bien que estuviéseris familiarizados con el concepto. En este documento explicamos brevemente que son los problemas interactivos y como funciona el juez: DMOJ.ca, donde están colgados los problemas de esta semana.

Puesto que en la IOI no está permitido usar Python, ninguno de los problemas de esta semana acepta soluciones usando Python.

Problemas interactivos

Los problemas interactivos son aquellos en los que deberéis crear un programa que interactúe con otro. Hay dos tipos de programas interactivos, aquellos que se comunican a través de funciones y aquellos que lo hacen a través del canal de entrada y salida:

Problemas interactivos a través de funciones

Este es el tipo de formato que tienen los problemas interactivos de la IOI y, por tanto, de los problemas de esta semana. En estos problemas nos piden que implementemos una función, la cual puede hacer llamadas a funciones que no se encuentran en el propio programa. El número de llamadas que podemos hacer a esas funciones suele estar restringido y, de hecho, la puntuación suele depender del número de llamadas (a menor número de llamadas mayor puntuación).

En este tipo de problemas es muy importante tener en cuenta que vuestro programa no puede tener main ni tener alguna función que se llame igual que las funciones a las que podéis llamar. Si no, vuestro código dará "Compilation Error".

Este tipo de problemas suele venir acompañado de una grader: un programa, que podéis ejecutar junto con el vuestro para probar casos. En los problemas interactivos de la IOI, se os darán diversos archivos, en particular uno de ellos se llamará grader.cpp y otro tendrá como nombre el nombre del problema. Si ponéis en este último fichero vuestra solución, podréis comprobar casos, ejecutando: `g++ grader.cpp problema.cpp` donde problema es el nombre del problema. Veréis que el fichero tiene por defecto unos includes: por ejemplo `#include "grader.h"` esos includes deben estar para que podáis ejecutar el grader. Pero **no deben estar presentes cuando enviéis el problema**.

Problemas interactivos a través del canal de entrada y salida

Este tipo de problemas aparecen en muchas plataformas, por ejemplo, en codeforces, pero no en la IOI. Aun así, os lo explicamos por si queréis practicar con problemas de otros sitios. Los problemas son muy parecidos a los anteriores, salvo que, en este caso, la comunicación se hace a través de la entrada y la salida. Vuestro programa deberá escribir por pantalla preguntas y el juez las leerá y escribirá, a su vez, una respuesta que deberéis leer.

En los problemas interactivos de este tipo, os tenéis que asegurar de que el output no se queda en vuestro buffer interno, y se envíe cuando debe ser enviado, para ello debéis usar la operación `flush` después de cada output. En c++ esto equivale simplemente a hacer `cout << endl` por lo que es importante no hacer los saltos de línea con `cout << '\n'`¹,

¹Este último comando hace lo mismo que endl, pero no vacía el buffer cada vez, por lo que no es conveniente usarlo en

en Python deberéis hacer `stdout.flush()` después de cada print.

Un ejemplo de problema de este tipo sería: <https://codeforces.com/problemset/problem/679/A>

DMOJ

DMOJ es una plataforma que contiene problemas de programación y en la que podéis enviar vuestro código para comprobar si es correcto o no. En particular, DMOJ tiene subidos todos los problemas de las olimpiadas internacionales desde el 2000, con lo cual es ideal para practicar.

Por lo que respecta a los problemas interactivos, DMOJ suele ofrecer un grader. Hay alguna excepción, en ese caso, nosotros os lo facilitaremos por Discord. Además, veréis que los enunciados están en inglés, también os enviaremos los enunciados en castellano.

problemas interactivos. No obstante en problemas normales, es una buena opción, ya que es más rápido que endl