**Preliminares**

*Si se sospechan intentos de trampa, el proceso de evaluación termina. Utiliza el flag de "Cheat" para reportarlo. Toma esta decisión con calma, inteligencia, y por favor, con cuidado.*

**Pruebas preliminares**

• La defensa sólo se puede realizar si el estudiante o grupo evaluado está presente. De esta manera, todos aprenden compartiendo conocimiento.

• Si no se ha entregado ningún trabajo (o el archivo, el directorio o el nombre del archivo es erróneo), la nota es 0 y la evaluacion se termina.

• Si alguno de los siguientes supuestos ocurre, la evaluación termina:

• El repositorio de Git está vacío.

• Hay fallo de Norma.

• Se detectan trampas, que resulta en -42.

• Existen fallos de compilación. El Makefile no debe hacer relink.

Si todos los puntos están bien, marca "Sí" y continúa la evaluación.  
En el caso contrario, usa el flag apropiado de acuerdo con las indicaciones  
al final de este scale.

 Yes

 No

**Instrucciones generales**

**Instrucciones generales**

• Si el programa principal muestra un error (segfault, bus error, sinsentidos, etc). Utiliza el flag de "Crash".

• El Makefile compila el ejecutable y tiene las normas requeridas.

• El nombre del ejecutable es "pipex".

• No se utilizan funciones prohibidas.

 Yes

 No

**Parte obligatoria**

*El comando "./pipex archivo1 comando1 comando2 archivo2" debe actuar como el comando "< archivo1 comando1 | comando2 archivo2".*

**Gestión de errores y de argumentos**

• El programa acepta exactamente 4 argumentos, ni más, ni menos (salvo para los bonus) y solamente en el orden correcto.

• La gestión de errores es correcta; archivos existentes, permisos de archivos, el binario del comando existe, etc.

Si todos los puntos se respetan, marca "Sí" y continúa la evaluación. En  
caso contrario, la evaluación recibe o "Incomplete work" o el flag que  
corresponda.

 Yes

 No

**El programa**

El programa hace lo que se le pide, sin mostrar más información/pasos  
que el comando shell que debe replicar.

Ejecuta tus propios tests y compara los resultados de shell al salir  
y el resultado de shell en comparación al del programa.  
Si no tienes ideas, mira los ejemplos del enunciado.

Si no se detectan errores, marca "Sí" y continúa. En caso contrario,  
la evaluación se termina aquí.

 Yes

 No

**Extras**

*Evalúa los bonus si, y solo si, la parte obligatoria está perfecta y completa; y la gestión de errores maneja comportamientos inesperados o de mal uso. En caso de que algún punto de la parte obligatoria no pasara la evaluación, toda la parte extra será completamente ignorada.*

**múltiples pipes**

El programa es capaz de gestionar el uso de múltiples pipes, uno tras  
otro. Para la parte obligatoria, prueba con pequeños comandos de shell  
y luego compara el resultado con el original.

Si funciona con dos pipes pero no con cinco, el bonus no cuenta.

 Yes

 No

**<< y >>**

El programa replica y permite el uso de << y >>. Prueba  
varias veces con cosas como : CMD << STOP\_VALUE | CMD1 >> file1

 Yes

 No