



Administration menu

- Accreditations
  - Cursus subscriptions
  - Dashes
  - Partnerships
  - Rules
  - Skills
  - Without mark teams
  - Project data
  - Transcripts
  - Quests
- Menu
- My projects
  - Holy Graph
  - List projects
  - Available Cursus

scale 6.1

Download

Index

Create new one

SCALE FOR PROJECT C PISCINE RUSH 02

Introduction

Par favor, respecta las siguientes indicaciones:

- Sé educado, cortés, respetuoso y constructivo durante el proceso de evaluación. El bienestar de la comunidad depende de ello.
- Identifica con la persona o grupo evaluado los posibles fallos del trabajo presentado. Tómate el tiempo de discutir y debatir los problemas que hayas identificado.
- Debes comprender que no todo el mundo entiende los enunciados del mismo modo, lo que quiere decir que vuestra visión del proyecto o lo que éste pide pueden variar. Mantén siempre una mente abierta y evalúa de la forma más sincera posible. El buen funcionamiento de la pedagogía se fundamenta en una evaluación correcta.

Guidelines

- No evalúes más que lo que figura en el repositorio git clonado del estudiante o del grupo.
- Asegúrate de que el repositorio git es efectivamente el del estudiante o el del grupo, que se trata del proyecto correcto, y que se ha lanzado « git clone » sobre un repositorio vacío.
- Asegúrate escrupulosamente de que no se ha utilizado para engañar ningún alias y de que vas a corregir la entrega oficial.
- Con el fin de evitar sorpresas, verifica con el evaluado los eventuales scripts usados para facilitar la evaluación.
- Si, como evaluador, no has realizado todavía el proyecto que vas a evaluar, debes leer el enunciado completamente antes de empezar a evaluar.
- Utiliza los flags disponibles para señalar una entrega vacía, un programa que no funcione, un error de norma, un problema de trampas... En esas situaciones, la evaluación se dará por finalizada y la nota será 0, o -42 si ha habido trampas. Sin embargo, salvo en los casos de trampas, te animamos a continuar con el intercambio de opiniones sobre el trabajo entregado, incluso no terminado, para identificar los problemas que han llevado al fracaso y evitar repetirlos en el futuro.
- A lo largo de toda la evaluación, no se tolerará ningún segfault ni ninguna parada del programa (inesperada, prematura o incontrolada) En esos casos, la nota final será 0. Utiliza el flag correspondiente.

No deberías tener que editar jamás ningún fichero, salvo el de configuración, si es que existe. En el caso en el que desees modificar un fichero, debes explicar explícitamente por qué necesita editarlo y ponerte de acuerdo con el estudiante evaluado antes de tocar cualquier cosa.

- Comprueba que sólo los archivos solicitados se encuentran en el repositorio git. De no ser así, la evaluación se dará por finalizada y la nota será 0

Attachments

- subject.pdf (translate)
- numbers.dict

Prólogo

Debes verificar la Norma incluso antes de probar el programa. Un error de Norma supone una falta de rigor y, como cualquier falta, la nota será un 0 en el rush. Cuando se tengan que evaluar ficheros y headers en C y solamente en C (ni makefile, ni script shell), solo será válida la Norminette.

El Makefile

El Makefile no debe hacer relink.

Si el Makefile vuelve a compilar cuando se hacen dos make seguidos, marque No. Si el Makefile no vuelve a compilar la segunda vez, marque Sí.

✓ Yes

✗ No

Gestión de errores

Asegúrate de que no haya archivos innecesarios en el repositorio (ocultos, temporales...). Aquí, debes identificar los errores potenciales del programa. ¿Cómo se comporta cuando le pasamos una cadena vacía? ¿Letras que no sean números? ¿Un número negativo? ¿Un número muy grande para un unsigned int? ¿Cómo se comporta el programa cuando le pasamos un diccionario como argumento? y sin el diccionario? Durante toda la evaluación, asegúrese de que no hay leaks en el programa.

✓ Yes

✗ No

Features testing

Comprueba los casos especiales y/o técnicos: 0, 00000, 10, 10000

Rate it from 0 (failed) through 5 (excellent)

Verificación general

Intenta averiguar cómo gestiona el programa los numero grandes. Prueba con números muy grandes - millares, millones, millardos... Concede puntos como sigue:

- Correcto hasta los millares: 2 puntos
- Correcto hasta los millones: 1 puntos
- Correcto hasta los millardos (menos que un unsigned integer positivo máximo): 1 punto
- Correcto por encima de un unsigned integer positivo máximo: 1 punto

Rate it from 0 (failed) through 5 (excellent)

Extras

Read

¿Gestiona el programa el syscall read sin argumento como parámetro?

✓ Yes

✗ No

Sintaxis correcta

¿Da el programa la respuesta correcta con la sintaxis correcta? (ej. 142 -> one hundred and forty-two)

✓ Yes

✗ No

Otro lenguaje

¿Hay otro lenguaje completamente funcional disponible (por ejemplo, con una opción)?

✓ Yes

✗ No

Ratings

Don't forget to check the flag corresponding to the defense

✓ Ok

★ Outstanding project

Empty work

No author file

Invalid compilation

Norme

Cheat

Crash

Incomplete group

Forbidden function

Conclusion

Leave a comment on this evaluation

Preview!!!

