



Piscina C

Rush 02

Resumen: Este documento corresponde a la evaluación del rush 02 de la piscina C de 42.

Índice general

I.	Instrucciones	2
II.	Los enunciados	4
III.	Extras	6

Capítulo I

Instrucciones


- Cualquier miembro del grupo puede inscribir el grupo al examen.
- El grupo debe estar inscrito al examen.
- Toda petición de precisiones sobre uno de los enunciados complicará los enunciados.
- Debe respetar el procedimiento de entrega para todos sus ejercicios.
- Los enunciados pueden cambiar hasta una hora antes de la entrega.
- La Moulinette compila con los flags -Wall -Wextra -Werror y utiliza cc.
- Si su programa no compila, tendrá 0.
- Debe entregar un Makefile, que compile su proyecto con reglas \$NAME clean y fclean
- Por lo tanto, deberá contestar a los enunciados indicados con los binomios impuestos y presentarse al examen a la hora acordada con todos sus binomios.
- El proyecto debe estar terminado cuando se presente al examen. Los exámenes sirven para presentar y explicar su trabajo hasta el último detalle.
- Cada miembro del grupo tendrá que estar perfectamente al corriente del trabajo realizado, se interrogará a cada uno de ellos, la nota del grupo se basará en las peores explicaciones.
- Evidentemente, tendrá que hacer todo lo posible para contactar con sus binomios: teléfono, e-mail, paloma mensajera, sesión de espiritismo, etc. No se aceptará ninguna excusa en lo que respecta a los problemas de grupo.
- Si después de haber intentado realmente todo no puede contactar con uno de sus binomios: realice su rush, nos las arreglaremos en el examen. Incluso si se trata del jefe del grupo: todos tienen acceso al repositorio.
- Obviamente, su trabajo deberá respetar la Norma: sea muy riguroso. Buen trabajo!



Es absolutamente obligatorio haber contestado perfectamente a los enunciados obligatorios para acceder a los enunciados extras.

Capítulo II

Los enunciados

	Ejercicio : 00
	rush-02
	Directorio de entrega : <i>ex00/</i>
	Ficheros a entregar : Makefile y todos los archivos necesarios
	Funciones autorizadas : write, malloc, free, open, read, close

- Debe realizar un programa que reciba un número como argumento de entrada y lo convierta en su valor escrito.
- Nombre del ejecutable: **rush-02**
- Su código fuente será compilado por el comando:

```
make fclean
make
```

- Su programa puede recibir hasta 2 argumentos:
 - Si solo hay un argumento, es el valor que necesita convertir.
 - Si hay dos argumentos, el primero es el nuevo diccionario de referencia y el segundo argumento es el valor que necesita convertir.
- Si el argumento no es un unsigned int, deberán reenviar 'Error' seguido de "\n".
- Por razones de armonización, su programa hablará en inglés.
- Su programa debe analizar el diccionario dado como recurso para el proyecto. Los valores dentro de el deben usarse para imprimir el resultado. Estos valores podrán ser modificados.
- Cualquier memoria asignada en la heap (con malloc(3)) debe liberarse correctamente. Esto se verificará durante la evaluación.

- El diccionario seguirá las siguientes reglas:

```
[a number][0 to n spaces]:[0 to n spaces][cualquier car\acter imprimible]\n
```

- Los números deben manejarse de la misma manera que atoi.
 - Debe recortar los espacios al principio y al final del valor en el diccionario.
 - El diccionario siempre tendrá al menos las claves dadas en el diccionario de referencia. Su valor puede ser modificado, se pueden agregar entradas pero no se pueden eliminar las claves iniciales.
 - Solo pueden utilizar los valores que se les ha dado inicialmente en el diccionario en anexos. (Por ejemplo, si añadimos la clave "54: fifty four", deberán utilizar siempre las mismas claves "50: fifty y 4: four")
 - Las entradas del diccionario pueden estar ordenadas con cualquier orden.
 - Pueden haber líneas vacías en el diccionario.
 - Si tienen algún error al analizar el diccionario, deberían lanzar "Dict Error\n". Su programa debe salir limpiamente.
 - Si el diccionario no les permite solucionar el valor demandado, lanzarán "Dict Error\n".
- Ejemplo:

```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
$> grep "20" numbers.dict | cat -e
20 :      hey      everybody !$
$> ./rush-02 20 | cat -e
hey everybody !$
```

Capítulo III

Extras

- Utilice datos personalizados de valor clave: `54: fifty-four` en vez de `50: fifty` y `4: four`
- Utilice `-`, `,`, `and` para estar más cerca de la sintaxis escrita correctamente
- Haciendo el mismo ejercicio con una lengua diferente. Para esto, se le permite proporcionar otro diccionario, el cual, contiene las entradas necesarias.
- Utilice `read` para leer la entrada estándar cuando no hay argumento.
- Cualquier otro extra que usted piense que aporta algo más al proyecto.