



Piscina C

Rush 00

Staff 42 piscine@42.fr

Resumen: Este documento corresponde al temario del rush 00 de la piscina C de 42.

Índice general

I.	Instrucciones	2
II.	Preámbulo	4
III.	Enunciados comunes	5
IV.	Rush 00	6
V.	Rush 01	8
VI.	Rush 02	9
VII.	Rush 03	10
VIII.	Rush 04	11

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.
- Los ejercicios han sido ordenados con mucha precisión del más sencillo al más complejo. En ningún caso prestaremos atención ni tendremos en cuenta un ejercicio complejo si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- La Moulinette compila con los flags -Wall -Wextra -Werror y utiliza cc.
- Si su programa no compila, tendrá 0.
- Los ejercicios del rush se deben realizar en grupos de 2, 3 ó 4.
- El número del rush impuesto a su grupo seguirá la siguiente regla: rango alfabético de la primera letra del login del team leader (de 1 a 26) módulo 5.
- 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 explicaciones menos buenas.
- 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.

- De manera opcional, puede contestar a varios enunciados para obtener posibles puntos extras.
- La Moulinette no tiene una mente muy abierta. No intenta comprender el código que no respeta la Norma. La Moulinette utiliza el programa `norminette` para comprobar la norma de sus archivos. Por lo tanto, entienda que es estúpido entregar un código que no pase la `norminette`.



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



Para esta jornada, la `norminette` debe ser ejecutada con el flag `-R CheckForbiddenSourceHeader`. La `moulinette` también lo utilizará.

Capítulo II

Preámbulo

He aquí la letra de los créditos de Pinky y Cerebro:

Pinky: Cerebro, ¿qué vamos a hacer esta noche?

Cerebro: Lo mismo que hacemos todas las noches, Pinky: ¡tratar de conquistar el mundo!

Son Pinky y Cerebro

Pinky y Cerebro

Uno es un genio

El otro no está cuerdo

De laboratorio son

Con genes insertados

Son Pinky

¡Son Pinky y Cerebro, bro, bro, bro,

bro, bro, bro, bro!

Antes del amanecer

Desarrollarán su plan

Y cuando salga el sol

El mundo conquistarán

Son Pinky y Cerebro

Pinky y Cerebro

Su motivación

Es fácil de explicar

Para probar su valor

El mundo conquistarán

Son Pinky


¡Son Pinky y Cerebro, bro, bro, bro,

bro, bro, bro, bro!

¡En lugar de conquistar el mundo, es preferible que se dedique a conquistar este rush!

Capítulo III

Enunciados comunes

	Ejercicio : 00
	rush0X
	Directorio de entrega : <i>ex00/</i>
	Ficheros a entregar : <i>main.c</i> , <i>ft_putchar.c</i> , <i>rush0X.c</i>
	Funciones autorizadas : write

- Los archivos que hay que entregar serán el *main.c*, un *ft_putchar.c* y su *rush0X.c*, donde 0X corresponderá al número del rush. Por ejemplo, *rush00.c*.
- Ejemplo de *main.c* :

```
int      main()
{
    rush(5, 5);
    return (0);
}
```

- Por lo tanto, deberá escribir la función **rush** que reciba como parámetro dos variables de tipo entero que se llamen respectivamente **x** e **y**.
- Su función **rush** tendrá que mostrar en pantalla un rectángulo de **x** caracteres de ancho e **y** caracteres de alto.
- Su **main** será modificado durante el examen para poder cambiar los parámetros de llamada a la función **rush**. Por ejemplo, se probarán este tipo de cosas:

```
int      main()
{
    rush(123, 42);
    return (0);
}
```

Capítulo IV

Rush 00

- `rush(5,3)` mostrará esto:

```
$> ./a.out
o---o
|   |
o---o
$>
```

- `rush(5, 1)` esto:

```
$> ./a.out
o---o
$>
```

- `rush(1, 1)` esto:

```
$> ./a.out
o
$>
```

- `rush(1, 5)` esto:

```
$> ./a.out
o
|
|
|
|
o
$>
```

- `rush(4, 4)` esto:

```
$> ./a.out  
o--o  
| |  
| |  
o--o  
$>
```


Capítulo V

Rush 01

- `rush(5,3)` mostrará esto:

```
$> ./a.out
/***\
*   *
\***/
$>
```

- `rush(5, 1)` esto:

```
$> ./a.out
/***\
$>
```

- `rush(1, 1)` esto:

```
$> ./a.out
/
$>
```

- `rush(1, 5)` esto:

```
$> ./a.out
/
*
*
*
\
$>
```

- `rush(4, 4)` esto:

```
$> ./a.out
/**\
*  *
*  *
\**/
$>
```

Capítulo VI

Rush 02

- `rush(5,3)` mostrará esto:

```
$> ./a.out
ABBBA
B  B
CBBBC
$>
```

- `rush(5, 1)` esto:

```
$> ./a.out
ABBBA
$>
```

- `rush(1, 1)` esto:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` esto:

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` esto:

```
$> ./a.out
ABBA
B  B
B  B
CBBBC
$>
```

Capítulo VII

Rush 03

- `rush(5,3)` mostrará esto:

```
$> ./a.out
ABBBC
B  B
ABBBC
$>
```

- `rush(5, 1)` esto:

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` esto:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` esto:

```
$> ./a.out
A
B
B
B
A
$>
```

- `rush(4, 4)` esto:

```
$> ./a.out
ABBC
B  B
B  B
ABBC
$>
```

Capítulo VIII

Rush 04

- `rush(5,3)` mostrará esto:

```
$> ./a.out
ABBBC
B  B
CBBBA
$>
```

- `rush(5, 1)` esto:

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` esto:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` esto:

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` esto:

```
$> ./a.out
ABBBC
B  B
B  B
CBBBA
$>
```