

Cuestionario

- (1 punto) 1. Indique las instrucciones en Bash para realizar un cambio a un subdirectorio denominado Direc y, si dicha acción ha tenido éxito, mostrar en pantalla los ficheros en dicho subdirectorio que tienen la extensión .c

Solución:

```
# Con un condicional:
if cd Direc; then
    ls *.c
fi
```

- (1 punto) 2. El programa grep, entre otras cosas, es capaz de buscar un texto en cada línea de un fichero de texto e imprime en la salida estándar cada una de esas líneas (ejemplo:

```
grep Sancho ElQuijote.txt
```

buscaría la palabra Sancho en el fichero ElQuijote.txt). El programa wc, entre otras cosas, cuenta el número de líneas en un fichero de texto, y si se invoca sin nombre de fichero, toma como fichero la entrada estándar (ejemplo para contar las líneas de la entrada estándar:

```
wc -l
```

Escribe una línea de comandos que cuente las apariciones de la palabra Sancho en el fichero ElQuijote.txt.

Solución:

```
grep Sancho ElQuijote.txt | wc -l
```

- (1 punto) 3. Supongamos que tenemos un programa *miprograma* en el directorio actual. Es habitual que para ejecutar dicho programa utilicemos el mandato

```
$ ./miprograma
```

¿Por qué es habitual?

A. Porque en general el directorio "." no está en el PATH

B. Por si acaso el fichero no tiene permiso de ejecución.

- (1 punto) 4. Si se ejecuta el mandato bash

```
$ ls -R /
```

aparecen por pantalla muchos ficheros y muchos mensajes de error. Indicar qué se debe añadir al mandato para que no aparezcan los mensajes de error.

Solución:

```
$ ls -R / 2> /dev/null
```

- (1 punto) 5. En un programa en language C se ha asignado memoria dinámica, en particular 2 espacios apuntados por los punteros p_int y p_array_ints.

```
p_int --> -----  
        | 4 |  
        -----  
  
p_array_ints --> -----  
                | 7 |  
                -----  
                | 3 |  
                -----  
                | 5 |  
                -----
```

Indique las instrucciones que liberen dicha memoria dinámica.

Solución:

```
free(p_int);  
free(p_array_ints);
```

- (1 punto) 6. Suponiendo que x es un vector de enteros y se ha ejecutado el siguiente código:

```
for (i=0; i<10; i++)  
x[i] = i+1;
```

Cuál es el valor de la expresión *x

Solución: 1

- (1 punto) 7. Escriba las sentencias necesarias para abrir y poder leer en un fichero llamado prueba.txt y posteriormente, solamente en el caso en que ha sido posible abrirlo, cerrar dicho fichero.

Solución:

```
FILE *fp;  
fp=fopen ("prueba.txt" , "r");  
if (fp != NULL ) fclose(fp);
```

- (1 punto) 8. Dado el siguiente código, escriba su salida tras su ejecución.

```
#include <stdio.h>  
int main ()
```

```
{
int var[] = {10, 100, 200};
int i, *ptr;

ptr = var;
printf("Valor_1=_ %d\n", *ptr );
ptr++;
printf("Valor_2=_ %d\n", *(ptr+1) );

return 0;
}
```

Solución: Valor 1 = 10

Valor 2 = 200

- (1 punto) 9. Dado la declaración del vector `int n[10]` , escribir las sentencias necesarias para inicializar cada uno de los elementos con un valor de 1.

Solución:

```
for ( i = 0; i < 10; i++ )
{
    n[ i ] = 1;
}
```

- (1 punto) 10. Sea el siguiente extracto de una función 'mifunc':

```
int mifunc(int n, float x) {
    /* codigo de la funcion */
}
```

Indique el prototipo (cabecera) de la función.

Solución:

```
int mifunc(int n, float x);
```

También

```
int mifunc(int, float);
```