

Cuestionario

El examen puntúa sobre 12 puntos. La duración del examen es 30 minutos.

En las preguntas con varias opciones, únicamente hay que contestar una. Las respuestas incorrectas no penalizan.

No se pueden tener hojas adicionales.

Parte de bash

- (1 punto) 1. En Unix, Indique cómo conectar *comando1* y *comando2* , de forma que el *comando2* únicamente se ejecute si el *exit status* de *comando1* es distinto de cero. .

Solución:

```
comando1 || comando2
```

- (1 punto) 2. En un script, indique los nombres de las variables que contienen el número de argumentos recibidos y el primer argumento.

Solución: #

```
1
```

- (1 punto) 3. Indique qué se muestra en pantalla (salida estándar) al ejecutar las siguientes instrucciones:

```
VAR1="uno_dos_tres"  
for ele in $VAR1  
do  
    echo $ele  
done
```

Solución: uno

```
dos
```

```
tres
```

- (1 punto) 4. 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
```

Parte de C

(1 punto) 5. Dadas las siguientes declaraciones y sentencia:

```
int a[5]={10,20,30,40,50}, *p=a, i=2;
printf("%d",*(p+i));
```

imprimirá:

- a) 20
- b) 30
- c) 40
- d) Nada al haber un error

Solución: b)

(1 punto) 6. Dado el siguiente programa,

```
main( )
{ int x=3;
  int *p=&x;
  /* Linea para imprimir el valor de x */
}
```

Indicar qué línea, de las siguientes, imprimiría correctamente el valor de x

- a) printf ("%d",*p);
- b) printf ("%d",p);
- c) printf ("%d",&p);
- d) printf ("%s",&p);

Solución: a)

(1 punto) 7. Sea el programa *examen* el resultado de compilar el código,

```
main(int argc, char const *argv[]){
    printf("%s", *(argv+1));
}
```

si ejecutamos `./examen hola adios` ¿Qué salida obtenemos?

Solución: hola

(1 punto) 8. Indique la salida del siguiente programa en C.

```
#include <stdio.h>
int main(void) {
    int a = 0;
    int *pa;
    a = 5;
    pa = &a;
    a = a+20;
    printf("%d", *pa);
    return 0;
}
```

- a) 5
- b) 25
- c) 0
- d) Fallo en la ejecución

Solución: b)

(1 punto) 9. Indique la salida del siguiente programa en C.

```
#include <stdio.h>
int main(void) {
    int i=0, r;
    while(i<3) {
        switch(i) {
            case 0:
                r = i;
                break;
            case 1:
                r = i+1;
                break;
            case 2:
                r = i+2;
            case 3:
                r = i+3;
        }
        i++;
    }
    printf("%i_", r);
    return 0;
}
```

- a) 0
- b) 4
- c) 5
- d) Fallo en la ejecución

Solución: c)

- (1 punto) 10. 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) 11. Dado el siguiente código,

```
int main (int argc, char*argv[])  
{  
    int A=1, B=2, C=3, *P1, *P2;  
    P1=&A;  
    P2=&C;  
    *P1=(*P2)+1;  
    printf("VALOR_DE_A_ %d_B_ %d_C_ %d_\n", A, B, C);  
}
```

indicar la salida de su ejecución:

Solución: VALOR DE A 4 B 2 C 3

- (1 punto) 12. Sean las siguientes declaraciones en lenguaje C:

```
struct nodo {  
    int n;  
    struct nodo *siguiente;  
};  
struct nodo *pnodo;
```

Apellidos:

Nombre:

Matrícula:

Indique las instrucciones C que consigan que pnode apunte a una estructura nodo asignada dinámicamente con los valores mostrados:

```
pnode ---->  -----  
              |    5    |  
              -----  
              |  NULL  |  
              -----
```

Solución:

```
pnode = (struct nodo *) malloc(sizeof(struct nodo));  
if (pnode == NULL) exit(1); /* opcional */  
pnode->n = 5;  
pnode->siguiente = NULL;
```