

# Examen

## Programación para Sistemas

Curso 2020/2021 - Enero 2021

### Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **media hora**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.
- No se pueden tener más que estas dos hojas en la mesa. Se puede usar parte de la segunda cara como borrador.
- En las preguntas que se ofrecen varias respuestas, sólo hay UNA correcta. NO penalizan las respuestas incorrectas.

### Cuestionario

- (1 punto) 1. Dar un mandato que recorra todo el sistema de ficheros mostrando su nombre en formato largo. Y que redirija la salida estándar al fichero todoslosficheros.txt y la salida de error al fichero errores .

**Solución:**

```
ls -alR / > todoslosficheros.txt 2> errores
```

- (1 punto) 2. Dar un mandato que cambie el directorio actual al directorio padre .

**Solución:**

```
cd ..
```

- (1 punto) 3. Asignar la salida estándar del mandato

```
date +%F
```

que hoy sería 2021-02-04 a una variable llamada hoy .

**Solución:**

```
hoy=$(date +%F)
```

- (1 punto) 4. Dar los mandatos necesarios para mover (renombrar) el fichero del directorio actual fichero.txt al subdirectorio del directorio actual destino solamente en el caso en que se den todas las condiciones para poder hacerlo-

**Solución:**

```
[ -r fichero.txt ] && [ -d destino ] && [ -w destino ]  
&& mv fichero.txt destino # escrito en una sola línea
```

- (1 punto) 5. ¿Cuál de las siguientes funciones es apropiada para solicitar memoria en C?
- A. new
  - B. malloc**
  - C. create
  - D. value
- (1 punto) 6. ¿Cuál de las siguientes expresiones devuelve la dirección de memoria de la variable `int a`?
- A. `*a`
  - B. `a`
  - C. `&a`**
  - D. `address(a)`
- (1 punto) 7. Escriba la salida que genera el siguiente programa en C :

```
#include <stdio.h>

int mi_funcion(int x)
{
    x=x*5;
    return x;
}

int main (int argc, char*argv[])
{
    int x=3;
    int y=3;
    y=mi_funcion(x);
    printf(" %d_ %d_ %d",mi_funcion(x),x,y);
}
```

**Solución:** 15 3 15

- (1 punto) 8. Dado el siguiente código C , que mostraría por la salida estándar

```
#include<stdio.h>
int main()
{
    char cadena[10] = {'H','o','l','a'};

    printf("%s",cadena);
    return 0;
}
```

**Solución:** Imprevisible. Probablemente la palabra Hola seguida de un montón de caracteres cualesquiera. En algún caso, puede fallar el programa.

- (1 punto) 9. Indique las sentencias necesarias en un programa C para abrir el fichero /home/mio/leer.txt en modo lectura. Y, además, compruebe que se ha podido abrir.

**Solución:**

```
#include<stdio.h>
int main()
{
    FILE *paa;

    if(( paa = fopen("/home/mio/leer.txt","r")) != NULL) /* sentencias,
        se ha abierto*/
    else /* sentencias, fallo al abrir*/
    return 0;
}
```

- (1 punto) 10. Escribe las funciones que se llamarían tras la ejecución del siguiente fragmento cuando c toma un valor de 3

```
switch (c) {
    case 1: Funcion1 ();
    case 2: Funcion2 ();
    case 3: Funcion3 ();
    case 4: Funcion4_1 ();Funcion4_2 ();break;
    case 5: Funcion_5 ();
    default: FuncionX ();
}
```

**Solución:** Funcion3, Funcion4.1 y Funcion4.2

- (1 punto) 11. Se pide escribir el comando necesario para compilar el código que se ha escrito en el fichero principal.c, si dicho fichero utiliza alguna función dada en auxiliar.c y auxiliar.h . El comando pedido debe generar un fichero ejecutable llamado: ejecutable\_examen

**Solución:** gcc principal.c auxiliar.c -o ejecutable\_examen

- (1 punto) 12. Usando typedef , definir un tipo que contenga por lo menos los mínimos elementos para trabajar con una lista enlazada (pila, cola, etc.) en la que en cada nodo se va a almacenar, al menos, un elemento de tipo int .

**Solución:**

```
typedef struct nodo{int n; struct nodo * p;} LISTA;
```