

Examen

105000119 - Programación para Sistemas 10MI-Grado en Matemáticas e Informática

Lenguajes y Sistemas Informáticos e Ingeniería de Software

ETSI Informáticos

Universidad Politécnica de Madrid

Curso 2018/2019 - Enero 2019

Normas

- El examen puntúa sobre **15 puntos**.
- La duración total del mismo es de **45 minutos**.
- 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 permite tener otro material que las tres hojas proporcionadas (se puede usar como borrador la última de las proporcionadas)
- En las preguntas en las que se ofrecen varias alternativas como posibles respuestas correctas, señalar una errónea no conlleva penalización

Cuestionario

- (1 punto) 1. En un script en Bash, indique las instrucciones de un condicional if-then-else necesarias para que, si el número de argumentos es distinto de 1, se acabe el script devolviendo un valor (exit status) de 1.

Solución:

```
if test $# -ne 1; then
    exit 1
fi
```

- (1 punto) 2. Suponiendo que se disponen los permisos para que la instrucción mkdir tenga éxito, indique la salida correspondiente a ejecutar las siguientes instrucciones en Bash:

```
mkdir NuevoDir && echo "A"
```

En pantalla aparece la salida: A

No se produce ninguna salida en pantalla

Ninguna de las anteriores

Solución: En pantalla aparece la salida: A

(1 punto) 3.Cuál debe ser la primera línea de un script ejecutable bash.

Solución:

```
#!/bin/bash
```

(1 punto) 4. Escriba un comando bash que, independientemente del directorio actual, le lleve a su directorio HOME

Solución: cd

(1 punto) 5. En Unix, indique la instrucción correspondiente (solamente una) para establecer el permiso de lectura, para todos los usuarios, de los ficheros uno.txt ejecutable.sh fichero.pdf del directorio actual

Solución: chmod a+r uno.txt ejecutable.sh fichero.pdf

Apellidos:

Nombre:

Matrícula:

- (1 punto) 6. En C indicar la forma de averiguar el tamaño de una cadena de caracteres (string) almacenada en `char c[2048]`

Solución: `strlen(c)`

- (1 punto) 7. Una aplicación en C se compone de dos ficheros fuente `miprog.c` (donde se encuentra el `main`) y `funciones.c`. Escriba la llamada al compilador `gcc` que compile los ficheros fuente y genere el ejecutable `miprog`.

Solución: `gcc -o miprog miprog.c funciones.c`

- (1 punto) 8. Un programa en C desea utilizar los argumentos proporcionados en línea de comandos. Indique la declaración de la función `main` de forma que se pueda acceder y utilizar el número de dichos argumentos y sus cadenas de caracteres (strings) correspondientes.

Solución:
`int main(int argc, char *argv[])`

- (1 punto) 9. Supóngase la siguiente declaración

`int a[N];`

Señala la expresión de C equivalente a `a[5]`¹

`*a+5`
`&a+5`
`*(a+5)`
`&(a+5)`

Solución:

`*(a+5)`

- (1 punto) 10. Dado el siguiente código C

```
char str1[] = "Hola";
char str2[] = "Hola";
if(str1 == str2)
    printf("Iguales\n");
else
    printf("Distintos\n");
```

¹el operador de suma `+` tiene menos prioridad que el operador de dereferencia `*`

Qué mostraría por la salida estándar?

A. Iguales

B. Distintos

C. Nada (daría un error).

(1 punto) 11. Dado el siguiente código C

```
int i = -1;
if (i) printf("Hola\n"); else printf("Adios\n");
```

Qué mostraría por la salida estándar?

A. Hola

B. Adios

C. Nada (daría un error).

(1 punto) 12. Dada la declaración de un vector n

```
int n[10], i, suma;
/* posteriormente se le asignan diversos valores a n[] */
```

Escribir el código necesario para calcular la suma de los elementos del vector n

Solución:

```
suma = 0;
for(i = 0; i < 10; i++) suma += n[i];
```

(1 punto) 13. Escriba las sentencias necesarias para declarar y reservar memoria dinámica para un vector de 325 números enteros, inicializando dicho vector con todas sus componentes a cero.

Solución:

```
int *vector;
vector = calloc( 325 , sizeof(int) );
```

(1 punto) 14. Escriba las sentencias necesarias para abrir y poder leer en un fichero llamado prueba.txt y posteriormente cerrar dicho fichero.

Solución:

```
file *fp;
fp=fopen ("prueba.txt" , "r");
fclose(fp);
```

(1 punto) 15. Se va a utilizar la siguiente declaración de doble puntero en lenguaje C:

```
char **ppchar;
```

la cual creará una variable tipo puntero:

```
ppchar
-----
|         |-->
```

Se desea llegar a obtener el siguiente diagrama:

```
ppchar
-----
|         |-->|         |-->| 'A' |
```

donde ppchar apunta a un puntero que a su vez apunta a un carácter al que se ha asignado el valor 'A'.

Complete el código siguiente para conseguirlo, debiéndose después liberar la memoria dinámica que se haya asignado antes de finalizar el programa.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {

    char **ppchar;
    /* Inicio del c\'odigo a completar */

    /* Fin del c\'odigo a completar */
    return 0;
}
```

Solución:

```
#include <stdio.h>
#include <stdlib.h>

int main( void ) {

    char **ppchar;

    ppchar = ( char ** ) malloc( sizeof( char * ) );
    if ( ppchar == NULL) { exit( 1 ); }

    *ppchar = ( char * ) malloc( sizeof( char ) );
```

```
if ( *ppchar == NULL) { exit( 1 ); }

**ppchar = 'A';

/* liberar la memoria din\amica asignada */
free( *ppchar );
free( ppchar );

return 0;
}
```