

Examen

105000016 - Programación para Sistemas Grado en Ingeniería Informática (2009)

Lenguajes y Sistemas Informáticos e Ingeniería de Software
Facultad de Informática
Universidad Politécnica de Madrid

Junio 2009/2010

Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **una 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.
- Las calificaciones se darán a conocer el **23 de junio** a través del Moodle de la asignatura.
- La revisión del examen tendrá lugar el **25 de junio** a las 16:00 en la sala 2319.

Cuestionario

(1 punto) 1. ¿Qué función debe tener todo programa en C ?

Solución: `main()`

(1 punto) 2. Indicar, si existe, un operador ternario en C.

Solución: `()? : ;`

(1 punto) 3. Si se desea realizar un bucle, ¿qué posibilidades nos ofrece C?

Solución: for , while , do while

(1 punto) 4. Escribir el resultado de aplicar el preprocesado del gcc al código:

```
#define AAA 10

#define CUADRADO(x) x*x

/* ... */

int i=1,b;

b = CUADRADO(AAA+i);
```

Solución:

```
int i=1,b;

b = 10 +i*10 +i;
```

(1 punto) 5. Sea la siguiente declaración de variable:

```
char palabra[] = "Hola";
```

¿Cuántos bytes en memoria ocupa el string `palabra`?

Solución: 5

(2 puntos) 6. Se está realizando un programa `prog` que tiene `prog.c` como fichero fuente asociado. El ejecutable ha dado un error de ejecución y se quiere llamar al depurador `gdb` con un `core` para intentar localizar dónde en el código se produce el error.

Indique *todas* las acciones, especificando las llamadas concretas a compilador, sistema operativo, depurador, etc., que debe realizar para ello.

Solución:

- Llamada al compilador con flag `-g`:
`gcc -g -Wall -ansi -pedantic prog.c -o prog`
- Llamada al comando `bash` `ulimit` para permitir la creación de ficheros `core`:
`ulimit -c unlimited`
- Llamada al ejecutable `prog` para crear el fichero `core` (tras error de ejecución):
`./prog`
- Llamada al depurador `gdb` con fichero `core` generado:
`gdb prog core`

(1 punto) 7. Sea el siguiente fragmento de código:

```
int misnumeros[] = {1, 2, 3, 4, 5};  
int *p;
```

```
p = &misnumeros[0];  
*p = *(p + 1) * (*p + 2);
```

¿Cuál es el contenido del array `misnumeros` después de la ejecución anterior?

Solución: {6, 2, 3, 4, 5}

Apellidos:

Nombre:

Matrícula:

(1 punto) 8. Indique el comando de Bash para obtener la fecha del día actual en el formato dd/mm/aa.

Solución: `date +%d/%m/%y`

(1 punto) 9. Indique el resultado que obtiene en pantalla al ejecutar los siguientes comandos

```
Uno=1; Dos=2
```

```
if (( Uno < Dos )); then echo S; else echo N; fi
```

Solución: S

```
if [[ Uno < Dos ]]; then echo S; else echo N; fi
```

Solución: N

(1 punto) 10. Indique el contenido de la variable cont tras ejecutar los comandos:

```
num=5; cont=0
```

```
until [[ num -eq 0 ]]; do { (( num-=1 )); (( cont+=num )); }; done
```

Solución: 10

```
num=5; cont=0
```

```
while [[ num -gt 0 ]]; do { (( cont+=num )); (( num-=1 )); }; done
```

Solución: 15

(1 punto) 11. Indique el contenido de las variables A, B y REPLAY tras ejecutar el comando:

```
read A B <<EOF
```

```
Uno Dos Tres
```

```
EOF
```

Solución:

- Contenido de A: Uno
- Contenido de B: Dos Tres
- Contenido de REPLAY: Null