

Extraordinaria-2024.pdf



superclash16



Programación



1º Grado en Ingeniería Electrónica Industrial



Escuela Superior de Ingeniería
Universidad de Almería



British
School of Languages



**Certifica tu inglés y
termina la carrera**

Certifica tu inglés con Aptis, online o presencial. Clases dinámicas y flexibles. **Matrícula ya disponible.** ¡Inscríbete!





la app para encontrar curro este verano
con más de 5000 ofertas de empleo.



descarga la app

Examen PROGRAMACIÓN (Grados en Ingeniería Mecánica, Eléctrica, Electrónica y Química Industrial) - Convocatoria Junio

Apellidos y nombre: _____ **Grupo:** _____

Instrucciones:

- El examen tiene dos partes. La primera son 20 preguntas tipo test que suponen 5 puntos sobre la nota final. La segunda, grapada al test, son dos ejercicios que suponen 3 puntos sobre la nota final. El examen completo se califica sobre 8 puntos.
 - Para seleccionar la respuesta correcta marcar con un círculo la opción, si es necesario anularla, tachar con un aspa. Si se vuelve a seleccionar una respuesta anulada marcarla como válida claramente con una flecha.
 - En todos los casos descritos se supone que los archivos .h han sido incluidos correctamente.
 - Cada pregunta tiene solo una opción correcta. Cada respuesta correcta suma **0.25** puntos. Cada tres respuestas erróneas anulan una correcta, es decir, cada respuesta incorrecta resta **-0.08** puntos.

1.- ¿Cuál completa la lista de unidades lógicas que forman una computadora?: Unidades de salida, de memoria, de control, aritmético lógica y de memoria secundaria.

2.- ¿Cuántos bits se necesitarían como mínimo para codificar un conjunto de 108 caracteres?

- (a) 4 (b) 5 (c) 6 (d) 7

3.- Indica a que elemento nos estamos refiriendo: "un dígito que puede asumir uno de dos valores y es el elemento de datos más pequeño de un ordenador".

- (a) Un número en binario
(c) Una posición de memoria
(b) Un bit
(d) No es nada de lo anterior

4 - ¿Cuál de las siguientes afirmaciones no es aplicable al hardware de una computadora?

- (a) El hardware de una computadora permite ejecutar programas de software.
 - (b) El hardware de una computadora consiste en componentes físicos como la CPU, la memoria y los dispositivos de entrada/salida.
 - (c) El hardware de una computadora es susceptible a ser reprogramado.
 - (d) El hardware de una computadora es responsable de ejecutar las instrucciones de un programa de software.

5 - ¿Cuál es la salida del siguiente programa?

```

int main()
{
    int i=3, j=2, k=0, m;
    m = ++i && ++j || ++k;
    printf("%d, %d, %d, %d\n", i, j, k, m);
    return 0;
}

```

itr 2 & i+j → T
 -3 & 3 → 1st augment, computer
 t+var → 1st augment, computer
 Var++ → 1st comp, 2nd augment
 NO -2 3 0 1

- 6.- ¿Cuáles de las siguientes afirmaciones son correctas sobre declaraciones if-else en un programa C? > 1. Cualquier declaración if-else puede ser reemplazada por una declaración equivalente utilizando el operador ternario ?: > 2. Las declaraciones if-else anidadas están permitidas > 3. Se permiten múltiples declaraciones en un bloque if. > 4. Se permiten múltiples declaraciones en un bloque else. > 5.

- (a) 1 y 2 (b) 2 y 3
(c) 1, 2 y 4 (d) 2, 3, 4

7.- ¿Cuál es el orden correcto en la expresión? $z = x + y * z / 4 \% 2 - 1$

$$F(0) = 1 \quad t+k$$

$t=T=m$



(a) $* / \% + - =$
(c) $/ * \% - + =$

(b) $= * / \% + -$
(d) $* \% / - + =$

8.- ¿Qué se muestra en pantalla?

```
int main(){
    int a=4, b=3, mayor;
    while ( (a<4) || (b<100) )
    {
        if (a>b){
            mayor = a;
        }else {
            mayor = b;
        } else { mayor = 0; }
        a=a+mayor;
        b=b*mayor;
    }
    printf("%i %i",a,b);
}
```

(a) 4 3 (b) 20 144
(c) 8 12 (d) Ninguna de las anteriores es correcta

9.- ¿Cuántas veces se ejecuta la instrucción printf?

```
int main() {
    int c = 1;
    while((c > 0) && (c < 30)) {
        if (c % 3) c++;
        printf("c = %d\n", c);
        c++;
    }
    return 0;
}
```

(a) 1 (b) 19 (c) 29 (d) 30

10.- ¿Qué imprime en pantalla el siguiente ejemplo?

```
int main() {
    int i, j, r;
    for(i=0; i<2; i++) {
        for(j=0; j<3; j++) {
            r = i+j;
            if(r % 2 == 0)
                printf("%i ", r);
        }
    }
    printf("\n");
    return 0;
}
```

(a) 0, 2, 4, 6 (b) 0, 2, 2
(c) 0, 2, 4 (d) No imprime nada

11.- ¿Cuál es la salida?

```
int function1(int);

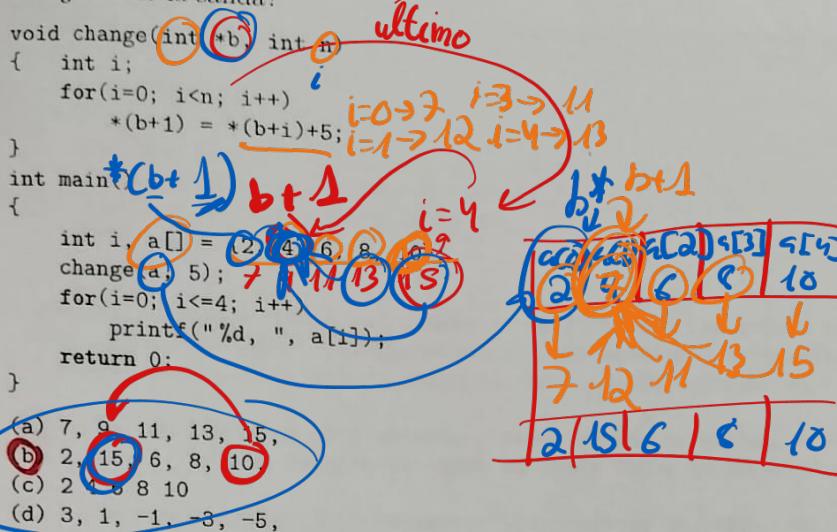
int main()
{
    int k=30;
    k = function1(k=function1(k=function1(k)));
    printf("k=%d\n", k);
    return 0;
}
int function1(int k)
{
    k++;
    return k;
}
```

(a) k=30

(b) k=31

(c) k=32

12.- ¿Cuál es la salida?



13.- Dado el siguiente programa escrito en C, decir el resultado en la consola:

```

void escribe(int i) {
    if(i < 0) return;
    escribe(-i);
    printf("%d ", i);
}

```

- (a) 0 1 2 3
(b) -1 0 1 2 3 4
(c) -1 0 1 2 3
(d) -1 0 1 2

14.- ¿Qué imprime el siguiente código?

```

int main()
{
    int i, a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    for(i=5; i>0; i--)
        printf("%d, ", a[2*i]);
    return 0;
}

```

- (a) 5, 4, 3, 2, 1, 0,
(b) 11, 9, 7, 5, 3,
(c) 10, 8, 6, 4, 2,
(d) 11, 9, 7, 5, 3, 1,

15.- ¿Cuál de las siguientes es no verdadera para los vectores en C?

- (a) Tienen un tamaño fijo una vez declarados
(b) Almacenan valores del mismo tipo
(c) Al usarlo de parámetro en funciones se modifica el array original
(d) Todas las anteriores son falsas

16.- De las siguientes opciones de declaración de nuevos tipos de datos, indique cuál seleccionaría para poder representar en C las frecuencias cardíacas medidas sobre un paciente desde las 9:00 hasta las 10:00 medidas cada 10 minutos: tipo latidos;

- (a) `typedef int tipo[1..7];`
(b) `typedef int tipo[0..6];`
(c) `typedef int tipo[7];`
(d) `typedef int tipo[6];`



la app para encontrar curro este verano
con más de 5000 ofertas de empleo.



descarga la app

descarga randstad app y empieza hoy.

17.- Indicar qué resultado se produce al ejecutar el siguiente programa:

```

typedef struct datos
{
    int i1, i2;
    double d;
    char c;
}tipo_registro;

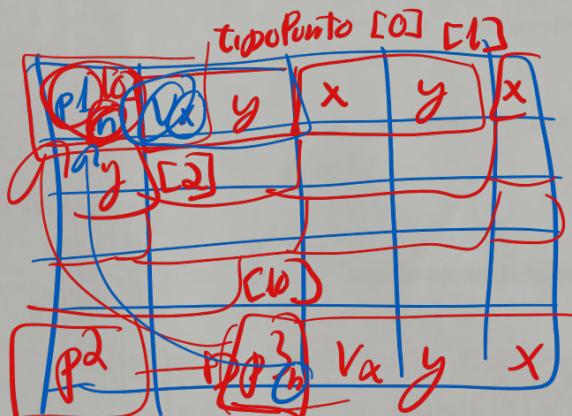
int main()
{
    tipo_registro dato1={2,10,5.55,'z'};
    tipo_registro *p=&dato1;
    p->c='a';
    printf("%d,%c,%c",p->i1,p->c,dato1.c);
}

```

(a) 2,a,z
(b) valor basura,a,valor basura
C 2,a,a
(d) valor basura.valor basura.z

- 18.- Dadas las siguientes estructuras y vectores, ¿qué código de los indicados producirá error?

```
typedef struct{
    double x,y;
} tipoPunto;
typedef tipoPunto tipoVector[10];
typedef struct{
    int n;
    tipoVector v;
} tipoLista;
int main(void){
    tipoLista p1, *p2;
    ...
}
(a) p1 = *p2;
(b) p1[0].x = 10; p1[0].y = 10;
(c) p1.n = (*p2).n;
(d) p1.v[0].x = 10; p1.v[0].y = 10;
```



19.- ;Qué hace el siguiente código?

```
int main() {
    FILE *archivo;
    char linea[256];
    archivo = fopen("nombre.txt", "r");
    fgets(linea, sizeof(linea), archivo);
    printf("%s", linea);
    fclose(archivo);
    return 0;
}
```

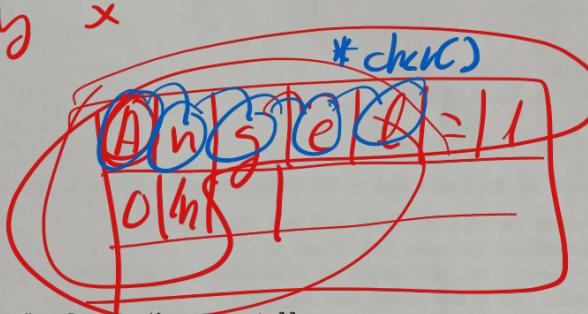
- (a) Produce un error de compilación

(b) Lee la primera línea del archivo "nombre.txt" y la escribe en pantalla.

(c) Se crea un archivo en disco llamado "nombre.txt".

(d) Ninguna de las anteriores es correcta

Write
figures



la escribe en pantalla.

Angel = 10 ↴ []

Chema = 10

nombreAngel char nombreAngel[10]
= nombreChim char nombreChim[10]



 randstad

20.- ¿Qué resultado producirá el siguiente fragmento de programa, suponiendo que en el origen de datos son todos correctos?

```

while (fscanf(ptreroFile, "%d %d %s %s %s", &id, &dni, apellido1, apellido2, nombre) != EOF)
{
    Alumnos[i].id=id;
    Alumnos[i].dni=dni;
    strcpy(Alumnos[i].apellido1, apellido1);
    strcpy(Alumnos[i].apellido2, apellido2);
    strcpy(Alumnos[i].nombre, nombre);
    i++;
}

```

a) Lee, en bucle, todas las líneas de un fichero y las guarda en un array de enteros y cadenas. Cada línea debe tener, obligatoriamente, dos enteros y tres cadenas de texto. Continúa hasta que se lee el carácter '\0' o hasta alcanzar el fin de archivo.

b) Lee cinco valores (dos enteros y tres cadenas de texto, en ese orden) y los guarda en un array de struct. Continúa leyendo de cinco en cinco hasta alcanzar el fin de archivo.

c) Lee cinco valores (dos enteros y tres cadenas de texto, en cualquier orden) y los guarda en un array de struct. Continúa leyendo de cinco en cinco hasta alcanzar el fin de archivo.

d) Produce error, puesto que en cada iteración del bucle debería utilizarse fopen() y fclose().

EJERCICIOS Ambos ejercicios suponen 1.5 puntos en la calificación final

2 dados

21.- Construir una función que muestra en pantalla todas las posibles tiradas de dados del monopoly. Es decir imprime el espacio muestral de tirar dos dados de seis caras, las tiradas de los dados vienen representadas por dos vectores que contienen los posibles valores de un dado. Realmente estamos imprimiendo el producto cartesiano de dos conjuntos representados sobre dos vectores de seis posiciones, es decir se imprimirán 36 pares de valores. No obstante la función "imprimir_espacio" debe ser valida para dos dados de cualquier número de caras siendo esto un parámetro pasado a la función.

```

#define DADO 6 // tamaño? ultimo valor? for
typedef int tipo_dado[DADO];

void imprimir_espacio(tipo_dado prim, tipo_dado segu, int n) {
    int cont=1, ind_val_prim, val_segu;
    for(ind_val_prim=0; ind_val_prim<n; ind_val_prim++)
        for(ind_val_segu=0; ind_val_segu<n; ind_val_segu++)
            printf("par número %d: dado 1 = %d, dado 2 = %d\n",
                   cont, val_prim, val_segu);
    cont++;
}

int main() {
    tipo_dado primero={1, 2, 3, 4, 5, 6};
    tipo_dado segundo={1, 2, 3, 4, 5, 6};

    imprimir_espacio(primer, segundo, DADO);

    return 0;
}

```

velores del dado

10
15