**Examen de Programación (Grados en Ingeniería Mecánica, Eléctrica, Electrónica y Química Industrial)**

**23 de Junio 2022**

**APELLIDO/S: NOMBRE:**

**DNI: GRUPO DOCENTE: GRUPO TRABAJO:**  **REPOSITORIO TI:**

**Instrucciones**:

Para seleccionar la respuesta (o respuestas) correcta(s), rodear la letra de la opción con un círculo **O**

Para anular una respuesta ya seleccionada tachar el círculo con una cruz **X**

Para volver a seleccionar de nuevo una respuesta anulada usar una flecha a la izq. de la opción **🡪**

El examen consta de 30 preguntas tipo test de respuestas múltiples

Cada pregunta admite una o varias opciones correctas (hay que seleccionar todas las opciones correctas)

Puntuación de cada pregunta: 8.0/30 (si se han seleccionado solamente todas las opciones correctas)

0.0 (en caso contrario)

* En cuál de los siguientes registros internos de la Unidad de Control de un procesador central (CPU) se encuentra la siguiente instrucción a ejecutar una vez que finalice la ejecución de la instrucción actual:

1. IR (“Instruction Record” 🡪 Registro de Instrucción)
2. PC (“Program Counter” 🡪 Contador de Programa)
3. SP (“Stack Pointer” 🡪 Puntero a la cima de la Pila)
4. En ninguno de los anteriores

* Cuál sería la representación binaria en complemento a dos del número -21, considerando un tamaño de representación de 6 dígitos para los números enteros con signo:

1. 110101
2. 101011
3. 101010
4. 010101
5. 001011
6. 001010
7. Ninguna de las anteriores

* En IEEE-754 simple precisión (32 bits, ne=8, nm=23), a qué número en el sistema de numeración decimal le corresponde la siguiente representación interna en hexadecimal: 7F8000001

1. 1.4\*10-45
2. -1.4\*10-45
3. 1.4\*10-38
4. -1.4\*10-38
5. +NaN
6. -NaN
7. +∞
8. -∞
9. Ninguno de los anteriores

* ¿Cuál o cuáles de las siguientes constantes literales son válidas en lenguaje C?

1. 0,00
2. 1.E-3
3. ' \\'
4. "Stop: '1' peligro"
5. 0xA73
6. "0xG73"
7. -1.0E4.2
8. true
9. '/n'
10. "/n"
11. Ninguna de las anteriores es válida

* Dado el siguiente código en lenguaje C para calcular la longitud del lado de un triángulo rectángulo opuesto al ángulo de 30°, dado por teclado la longitud de la hipotenusa, señale la afirmación o afirmaciones correctas:

|  |  |
| --- | --- |
| * 1. El programa calcula e imprime correctamente la longitud del cateto correspondiente   2. Error de compilación: falta la librería **stdlib.h**   3. Error de compilación: falta un ; en la tercera línea (tras 30.0)   4. Error de diseño en el cálculo de a: debería ser a=h/sin(ANG);   5. Todas las afirmaciones anteriores son incorrectas | #include <stdio.h>  #include <math.h>  #define ANG 30.0 /\* 30 grados \*/  int main(){  float a,h;  printf("Introduzca hipotenusa: ");  scanf(" %f", &h);  a=h\*sin(ANG);  printf("a: %.2f", a);  return 0;  } |

* En C, si tenemos que **a**, **b** y **c** son variables de tipo **int**, **float** y **char**, respectivamente, y que tienen asignados los valores a=1, b=5.1 y c='x', indicar qué expresión (o expresiones) devuelven como resultado verdadero (distinto de cero):
  1. !((a>b)||(c!='x'))
  2. (a==(a=3)%2) || ! (a && c)
  3. --a || ! c && b
  4. (a<b)&&(a>-b)&&(!a)
  5. Ninguna de las anteriores
* ¿Qué mensaje aparece en pantalla al ejecutar el siguiente programa en C?

|  |  |
| --- | --- |
| * 1. Temperatura=0 grados   2. Temperatura positiva   3. Temperatura negativa   4. (No aparece nada en pantalla)   5. Error de compilación   6. Error de ejecución   7. Ninguna de las respuestas anteriores es correcta | #include <stdio.h>  int main(){  double temp=-30.0;  if(temp=0)  printf("Temperatura=0 grados");  else if(temp>0)  printf("Temperatura positiva");  else if(temp<0)  printf("Temperatura negativa");  return 0;  } |

* ¿Qué mensaje se presenta en pantalla cuando se ejecuta el siguiente programa?

|  |  |
| --- | --- |
| 1. 0 2. 1 3. 2 4. 3 5. 4 6. (No aparece nada en pantalla) 7. Error de compilación 8. Error de ejecución 9. Ninguna de las respuestas anteriores es correcta | #include <stdio.h>  int main(){  int a=2;  switch(a){  case 1: a++;  case 2: ++a;  case 3: a++;  default: --a;  }  printf("%d",a--);  return 0;  } |

* Indicar los resultados que imprime en pantalla el siguiente programa en C:

|  |  |
| --- | --- |
| 1. Suma=0 2. Suma=10 3. Suma=55 4. Suma=110 5. Error de compilación 6. Error de ejecución (bucle infinito) 7. Ninguna de las respuestas anteriores es correcta | #include <stdio.h>  int main(){  int i=0, n=10, suma=0;  for(;i<=10;++i);  suma+=i;  printf("Suma=%d", suma);  return 0;  } |

* Tras ejecutar el siguiente programa en C, ¿qué valores se muestran por pantalla?

|  |  |
| --- | --- |
| * 1. 3600=1   2. 3600=1\*2   3. 3600=1\*2\*2\*2\*3\*5   4. 3600=1\*2\*2\*2\*2\*3\*3\*5\*5   5. 3600=1\*2\*2\*2\*2\*2\*3\*3\*3\*5\*5   6. Error de compilación   7. Error en tiempo de ejecución   8. Ninguna de las respuestas anteriores es correcta | #include <stdio.h>  int main(){  int n=3600,i=2;  printf("%d=1",n);  while((n!=1)&&(i<=n))  if(n%i) ++i;  else{ n/=i;  printf("\*%d",i);  }  return 0;  } |

* ¿Qué mensaje se presenta en pantalla cuando se ejecuta el siguiente programa?

|  |  |
| --- | --- |
| * 1. El valor de x es 50   2. El valor de x es 55   3. El valor de x es 60   4. El valor de x es 65   5. El valor de x es 100   6. El valor de x es 102   7. (No aparece nada en pantalla)   8. Error de compilación   9. Error de ejecución (bucle infinito) | #include <stdio.h>  int main(void) {  float x=0.0;  int i,j,k;  for(i=1;i<=5;i++)  for(j=1;j<=i;j++)  for(k=j;k<=5;k++)  x++;  printf("El valor de x es %.0f",x);  return 0;  } |

* ¿Cuál (o cuáles) de las siguientes afirmaciones es cierta en relación con el prototipo de una función en C?

1. Es opcional su inclusión al principio del código fuente
2. Si se incluyen, el orden de escritura de las definiciones de las funciones en el código fuente es arbitrario.
3. El prototipo de una función coincide con la primera línea de definición de una función, pero acabada en ;
4. No existen los prototipos de funciones en C en programas que constan de una única función.
5. Ninguna de las respuestas anteriores es cierta

* ¿Cuál es el resultado que se muestra por pantalla tras ejecutar el siguiente programa?

|  |  |
| --- | --- |
| * 1. x=3 y=5   2. x=5 y=3   3. x=3.000000 y=5.000000   4. x=5.000000 y=3.000000   5. Error de compilación   6. Error en tiempo de ejecución   7. Ninguna de las respuestas anteriores | #include <stdio.h>  void intercambiar(double \*x, double \*y);  int main(void){  double x=3, y=5;  intercambiar(x,y);  printf("x=%.0f y=%.0f",x,y);  return 0;  }  void intercambiar(double \*x, double \*y){  \*x=\*y-\*x;  \*y=\*y-\*x;  \*x=\*x+\*y;  } |

* Dada el siguiente programa en C, indicar el resultado que aparecería en pantalla tras su ejecución:

|  |  |
| --- | --- |
| 1. Hola   Hola  Hola   1. 3   Hola   1. 2   Hola   1. Hola   3   1. Hola   2   1. Ninguna de las anteriores | #include <stdio.h>  int saludos(int n);  int main(){  printf("%d",saludos(3));  return(0);  }  int saludos(int n){  printf("Hola\n");  return --n;  } |

* Dado el siguiente fragmento de programa: ¿cuál o cuáles de las activaciones de la función **f** podría escribirse en la línea de guiones (-----------) para que el programa compilase sin errores ni avisos (“warnings”)?

|  |  |
| --- | --- |
| 1. f(x,c,&z); 2. z=f(\*x,\*y,c); 3. f(4,z,&x); 4. f(4,'A',z); 5. f(4,'A',3.3); 6. Error de compilación 7. Todas las anteriores son incorrectas. | #include <stdio.h>  void f(int x, char c, float \*z);  int main(){  int c=4;  char z='C';  float x;  -------------------  return(0);  }  void f(int x, char c, float \*z){  /\* Definición de función \*/  } |

* Considerar el siguiente programa en C, donde la función llamada hace uso de la variable global (o externa) **a**. Después de la ejecución de este programa, ¿cuál es el resultado que aparecerá en pantalla?

|  |  |
| --- | --- |
| 1. Variable a = 0 2. Variable a = 1 3. Variable a = 2 4. Variable a = 3 5. Variable a = 4 6. Variable a = NaN 7. Error de compilación 8. Todas las respuestas anteriores son incorrectas | #include <stdio.h>  int a=1;  void proc (int \*b, int \*c){  \*b=3;  \*c=a%2;  \*b=a+(\*b)\*(\*c);  }  int main(){  proc(&a,&a);  printf("Variable a = %d", a);  return 0;  } |

* Indicar qué escribe en pantalla el siguiente programa al ejecutarlo:

|  |  |
| --- | --- |
| 1. 13 2. 0 3. 1 4. 1 1 0 1 5. 1 0 1 1 6. Error de compilación 7. Error en tiempo de ejecución | #include <stdio.h>  void imprime\_binario(int n){  imprime\_binario(n/2);  printf(" %d" , n%2);  }  int main(){  imprime\_binario(13);  return 0;  } |

* Considere el siguiente prototipo de una función C:

double f(double (\*f)(double x,double y),double x, double y);

¿Cuál o cuáles de las siguientes instrucciones son correctas?

1. printf(" Resultado= %f" ,f(sin,M\_PI,2\*M\_PI));
2. printf(" Longitud= %f" ,f(atan,5.0,4.0));
3. printf(" Longitud= %f" ,f(pow,5.0,4.0));
4. printf(" Longitud= %f" ,f(pow+sin,5.0,4.0));
5. Ninguna de las anteriores (error de sintaxis en el prototipo de la función)

* ¿Cuál (o cuáles) de las siguientes instrucciones permiten declarar en C una variable “array” unidimensional de 100 reales en doble precisión?
  1. typedef double v[100];
  2. double v[100];
  3. typedef tipo\_vector double[100];

tipo\_vector v;

* 1. typedef double tipo\_vector[100];

tipo\_vector v[100];

* 1. typedef double tipo\_vector;

tipo\_vector v[100];

* 1. typedef double tipo\_vector[100];

tipo\_vector v;

* 1. Ninguna de las anteriores
* Considerar el siguiente programa en C, e indicar qué resultado(s) aparecería en pantalla tras su ejecución:

|  |  |
| --- | --- |
| * 1. 1 -3 -5 2 8 -7 4   2. 1 -3 2 8 4   3. 1 -5 2 8 -7 4   4. 1 -5 2 8 4   5. 1 2 8 -7 4   6. 1 2 8 4   7. Error de sintaxis (faltan llaves {})   8. Error en tiempo de ejecución   9. Ninguna de las anteriores | #include <stdio.h>  #define MAX 100  typedef int tipo\_vector[MAX];  int main(){  int i,j,n;  tipo\_vector v={1,-3,-5,2,8,-7,4};  n=7;  for(i=0;i<n;++i){  if(v[i]<0){  for(j=i;j<n-1;++j)  v[j]=v[j+1];  n--;  }  }  for(i=0;i<n;++i)  printf("%d ",v[i]);  return 0;  } |

* De las siguientes declaraciones de nuevos tipos de datos en C, indicar cuál o cuáles seleccionaría para representar los valores de la temperatura atmosférica terrestre, medidas en múltiplos puntos con las siguientes características:
  + - Todos ellos situados a 10000 km en dirección radial medidos desde el centro de la tierra.
    - Separados cenitalmente por 10° de latitud desde -90° hasta +90°.
    - Separados acimutalmente por 10° de longitud desde 0° hasta +350°.

1. typedef double tipo[19];
2. typedef double tipo[36];
3. typedef double tipo[19][36];
4. typedef double tipo[36][19];
5. typedef double tipo[10][19][36];
6. typedef double tipo[10][36][19];
7. Ninguna de las anteriores
8. ¿Cuál o cuáles de los siguientes fragmentos de código en C utilizaría para trasponer una matriz dada como parámetro de E y S?

|  |  |
| --- | --- |
| * 1. for(i=0;i<\*f;++i)   for(j=0;j<\*c;++j){  x=A[i][j];  A[i][j]=A[j][i];  A[j][i]=x; }   * 1. for(i=0;i<\*c;++i)   for(j=0;j<\*f;++j){  x=A[i][j];  A[i][j]=A[j][i];  A[j][i]=x; }   * 1. for(i=0;i<\*f;++i)   for(j=i+1;j<\*c;++j){  x=A[i][j];  A[i][j]=A[j][i];  A[j][i]=x; }   * 1. for(i=0;i<\*c;++i)   for(j=i+1;j<\*f;++j){  x=A[i][j];  A[i][j]=A[j][i];  A[j][i]=x; }   * 1. Ninguno de las anteriores | #include <stdio.h>  #define FIL 10  #define COL 10  typedef double array2d[FIL][COL];  void traspuesta(array2d A, int \*f, int \*c){  int i,j;  double x;  i=\*f;  \*f=\*c;  \*c=i;  /\* CODIGO A INSERTAR \*/  }  int main(){  array2d A={{1,2,3,4},{5,6,7,8}};  int f=2, c=4;  traspuesta(A,&f,&c);  … |

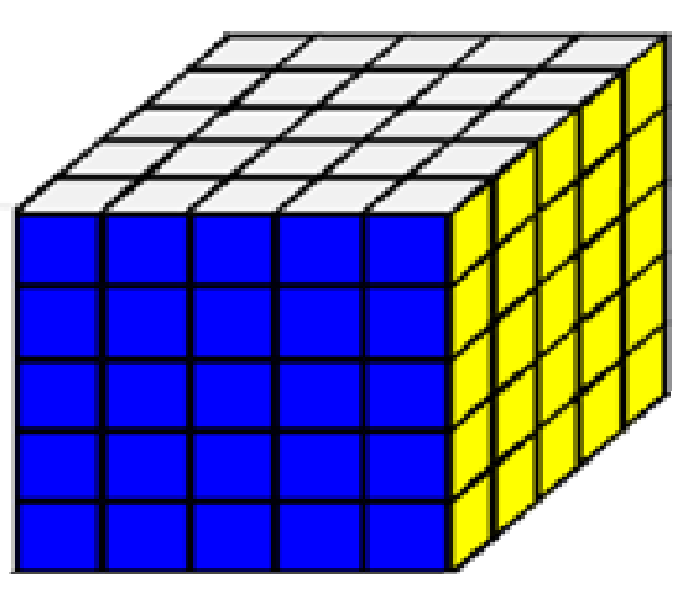
* Indicar cuál o cuáles de los siguientes fragmentos de código es el adecuado para que la correspondiente función en C elimine todos los caracteres no alfabéticos de la cadena de caracteres que se le pasa como argumento:

|  |  |
| --- | --- |
| 1. while(cad[i]){   if(isalpha(cad[i])){  cad[j]=cad[i];  ++j;  }  i++;  }   1. while(cad[i]!='\0'){   if(isalpha(cad[i])=='1'){  cad[i]=cad[j];  ++j;  }  i++;  }   1. for(i=0;i!='\0';++i){   if(isalpha(cad[i])){  cad[j]=cad[i];  ++i;  }  j++;  }   1. for(i=0;i!='\0';++i){   if(isalpha(cad[i])){  cad[j]=cad[i];  ++j;  }  i++;  }   1. Ninguna de las anteriores | #include <stdio.h>  #include <ctype.h>  int eliminar(char \*cad){  int i=0,j=0;  /\* FRAGMENTO DE CODIGO \*/  cad[j]='\0';  }  int main(){  static char cad[30]="Hola 12.;adios+3 ";  eliminar(cad);  printf("%s",cad);  return 0;  } |

* Indicar cuál o cuáles de las siguientes sentencias en relación con las estructuras de dato tipo registro en lenguaje C (**struct**), son correctas:
  1. No son estructuras de datos estáticas.
  2. Solo pueden contener componentes que obligatoriamente tienen que ser de tipos de datos diferentes.
  3. Los componentes de un registro tienen que ser datos simples.
  4. El acceso a los componentes se realiza a través de su nombre o identificador mediante el operador de indexación [].
  5. El acceso a los componentes se realiza a través de su posición relativa mediante el operador de selección de campo ->.
  6. El acceso a los componentes se realiza a través de su nombre o identificador mediante el operador de selección de campo ->.
  7. Al igual que los “arrays”, solo se pueden pasar a través de las interfaces de las funciones por dirección.
  8. Ninguna de las respuestas anteriores es correcta
* Dado el siguiente fragmento de código en lenguaje C indique cuál o cuáles de los siguientes bloques de instrucciones son correctos (tanto sintáctica como semánticamente):

|  |  |
| --- | --- |
| * 1. p2=p1;   2. p2=\*p1;   3. &p1=p2;   4. p2=&p1;   v=p2->v;   * 1. p1.v.x=75.0;   p1.v.y=45.3;   * 1. p1[0].v.x=100;   p1[0].v.y=100);   * 1. p2=&p1;   p2.v[0]->x=100;  p2.v[0]->y=200;   * 1. Ninguno de los anteriores | #include <stdio.h>  #include <math.h>  typedef struct{  double x,y;  }tipPunto;  typedef tipoPunto tipoVector[10];  typedef struct{  int n;  tipoVector v;  }tipoLista;  int main(void){  tipo\_lista p1,\*p2;  tipo\_vector v;  p1.n=3;  ...  return 0;  } |

* De las siguientes declaraciones de nuevos tipos de datos, indicar cuál o cuáles seleccionaría para representar únicamente los códigos de color (0: Rojo, 1: Verde, 2: Azul,…) de las caras visibles de cada pieza de un cubo de Rubik formado por N3 cubos unitarios:



* 1. typedef int tipoPieza[6];

typedef tipoPieza tipoCubo[N][N][N];

* 1. typedef int tipoCubo[6][N][N];
  2. typedef int tipoCara[6][6];

typedef tipoCara tipoCubo[N];

* 1. typedef int tipoCara[N][N];

typedef struct{

tipoCara sup,inf,izq,der,del,det;

}tipoCubo;

* 1. Ninguna de las anteriores
* ¿Cuántas iteraciones requiere el bucle del mejor algoritmo de búsqueda secuencial para localizar la posición de la primera ocurrencia de un elemento de clave=2 en el vector V?

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V | 9 | 9 | 8 | 7 | 7 | 7 | 6 | 6 | 5 | 5 | 4 | 3 | 1 | 1 | 1 | 1 | 0 |

1. 12
2. 13
3. 14
4. 15
5. 16
6. 17
7. No se puede aplicar el algoritmo de búsqueda secuencial a este vector

* Considerando la siguiente declaración: **FILE \*f;**, indicar qué instrucción (o instrucciones) pueden ser correctas para crear y abrir para escritura un archivo de texto de nombre “fichero.txt” situado en el directorio actual:

1. fopen(f, "fichero.txt", "w");
2. fopen(f, "fichero.txt","a");
3. f==fopen("fichero.txt","w");
4. f==fopen("fichero.txt","a");
5. f=fopen("fichero.txt",'w');
6. f=fopen("fichero.txt", 'a');
7. fopen("fichero.txt", 'w');
8. fopen("fichero.txt", 'a');
9. Ninguna de las anteriores

* Tenemos un archivo de texto en el directorio actual llamado “sensores.txt” conteniendo cada línea 10 números reales separados por al menos un espacio en blanco. Considerando las siguientes declaraciones y suponiendo que el archivo se ha abierto correctamente para lectura a través de la variable f, si queremos leer una línea completa del archivo, indicar qué instrucción o instrucciones son las correctas:

1. fscanf(f, " %10f", v);

#include <stdio.h>

typedef float tipo\_vector[10];

int main(){

FILE \*f;

int i;

tipo\_vector v;

. . .

return 0;

}

1. fscanf(f, " 10%f", v);
2. for(i=0;i<10;++i)

fscanf(f, " %f", v[i]);

1. for(i=1;i<=10;++i)

fscanf(f, " %f",&v[i]);

1. i=0;

while(!feof(f)

fscanf(f, " %f", &v[i++]);

1. Ninguna de las anteriores es correcta

* Tenemos un archivo de texto en el directorio actual llamado “notas.txt” conteniendo la información de todos los alumnos de una asignatura con el siguiente formato:

Nombre: Juan Apellido: García Asignatura: Programación Nota: 6.66

Nombre: Pedro Apellido: Revilla Asignatura: Programación Nota: 4.83

Nombre: Arturo Apellido: Rodríguez Asignatura: Programación Nota: 8.15

…

Si queremos leer una línea del mismo, suponiendo que el archivo se ha abierto para lectura correctamente a través de la variable f, y dadas las siguientes variables, indicar qué instrucción o instrucciones son las correctas:

char nombre[50], apellido[50], asignatura[50];

FILE \*f;

float nota;

1. fscanf(f, " Nombre: %s Apellido: %s Asignatura: %s Nota: %f", nombre, apellido, asignatura, nota);
2. fscanf(f, " Nombre: %s Apellido: %s Asignatura: %s Nota: %f", &nombre, &apellido, &asignatura, &nota);
3. fscanf(f, " Nombre: %s Apellido: %s Asignatura: %s Nota: %f", nombre, apellido, asignatura, &nota);
4. fscanf(f, " Nombre: %s Apellido: %s Asignatura: %s Nota: %f", &nombre, &apellido, &asignatura, &nota);
5. Ninguna de las anteriores