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

**24 de Mayo 2020**

**APELLIDO/S: NOMBRE:**

**DNI: GRUPO DOCENTE: GRUPO TRABAJO:**

**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)

* ¿Cuál ó cuáles de las siguientes afirmaciones son correctas?

1. La representación en C de un número con el tipo float es exacta pero con límites
2. La representación en C de un número con el tipo int es exacta pero con límites
3. La representación en C de un número con el tipo float es exacta y sin límites
4. La representación en C de un número con el tipo int es exacta y sin límites
5. Todas las afirmaciones anteriores son incorrectas

* Respecto a los nombres o identificadores en lenguaje C, marcar la opción u opciones correctas:

1. Puede contener espacios en blanco
2. Puede ser de cualquier longitud
3. Pueden contener cualquier secuencia de letras, números y caracteres de subrayado (\_)
4. No se diferencia entre letras mayúsculas y minúsculas
5. Ninguna de las otras opciones es correcta

* Dado el siguiente código en lenguaje C, señale la afirmación o afirmaciones correctas:

|  |  |
| --- | --- |
| * 1. El programa tiene un error que se detecta en tiempo de compilación (no compila).   2. La variable **res** almacena el doble del valor de **num**.   3. El programa muestra en pantalla el valor acumulado en la variable **res**.   4. El programa muestra por pantalla la dirección de la variable **res**.   5. Todas las respuestas anteriores son incorrectas | #include <stdio.h>  int main(void){  float num, \*res;  printf ("Introduzca un numero:\n");  scanf(" %f", &num);  res = 2\*num;  printf("%f", \*res);  return 0;  } |

* Indique cuál (o cuáles) de las siguientes afirmaciones es verdadera en lenguaje C:
  1. Una variable de tipo "int" puede almacenar valores negativos.
  2. Una variable de tipo "float" no puede almacenar valores enteros.
  3. La sentencia “char letra=b;” asigna a la variable **letra** el valor correspondiente a la letra b.
  4. La sentencia float f=3,2 asigna a la variable **f** el valor **3,2**.
  5. Todas las afirmaciones anteriores son verdaderas
  6. Todas las afirmaciones anteriores son falsas
* En lenguaje C ¿qué operación (u operaciones) de las siguientes es correcta?:
  1. 25 \* 3,5
  2. 33.7 % 5.0
  3. 33 / 5.3
  4. 10,5 % -15,4E2
  5. Todas las operaciones anteriores son incorrectas
* En lenguaje C, ¿cuál o cuáles de las siguientes afirmaciones es correcta respecto del operador binario %?
  1. Puede dar error para operandos con valores naturales (1,2,3,…)
  2. Nunca da error para operandos de tipo int
  3. Nunca da error para operandos de tipo float, siempre que el operando de la derecha sea no nulo
  4. Está definido para operandos de cualquier tipo de datos
  5. No existe este operador en C
  6. Todas las respuestas anteriores son falsas
* En C, si tenemos tres variables enteras con los siguientes valores asignados: **a=1**, **b=0** y **c=0**, indicar ¿qué expresión (o expresiones) devuelven como resultado **1** (verdadero)?:
  1. !(! a || c || !b)
  2. !(! c || b && a)
  3. a && ! (a || ! c)
  4. a && ! c || b
  5. Ninguna de las anteriores devuelve 1
* Dado el siguiente fragmento de código en C, Después de su ejecución la variable **n** contiene el valor:

|  |  |
| --- | --- |
| * 1. -1   2. 0   3. 1   4. 2   5. 3 | int n=0;  if(n=2){  n=n+1;  }else{  n=n-1;  } |

* En lenguaje C, ¿cuál ó cuales de las siguientes sintaxis es correcta para el bucle **for** si **n** es una variable de tipo **int**?:

1. for(n,n<3,n++)
2. for( ;n,n++)
3. for(;;)
4. for(;n<3;n+)
5. Todas las anteriores son incorrectas

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

|  |  |
| --- | --- |
| * 1. 4 3   2. 20 144   3. 8 12   4. Error de compilación   5. Ninguna de las respuestas anteriores es correcta | #include <stdio.h>  int main(void){  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);  } |

* ¿Qué imprime el siguiente programa en pantalla?

|  |  |
| --- | --- |
| * 1. 6   2. 5   3. 4   4. 3   5. No imprime nada (bucle infinito) | #include <stdio.h>  int main(void) {  int i=0, r;  while(i<3){  switch(i) {  case 0: r = i;  break;  case 1: r = i+1;  break;  case 2: r = i+2;  case 3: r = i+3;  }  i++;  }  printf("%i ", r);  return 0;  } |

* ¿Es posible ejecutar un programa construido en C que no tenga la función **main()**?

1. No, en ningún caso
2. Sí, siempre es posible
3. Sólo en el caso de que el programa incluya la librería <stdio.h>
4. Sólo en el caso de que el programa tenga otra función

* La palabra reservada **void** en la declaración de la función: **void func(int a, float \*b);** indica que…
  1. La función no devuelve ningún resultado a través de la lista de parámetros formales.
  2. La función devuelve el valor 0.0 a través del nombre o identificador de la misma.
  3. Todos los parámetros formales pueden ser excepcionalmente de salida.
  4. No indica nada (void significa nulo).
  5. Es un error de compilación.
  6. Ninguna de las respuestas anteriores es correcta.
* ¿Cuál es el resultado que se muestra por pantalla tras ejecutar el siguiente programa?

|  |  |
| --- | --- |
| * 1. 140, 5   2. 140, 20   3. 35, 5   4. 35, 20   5. Error de compilación | #include <stdio.h>  int producto(int a,int b);  int main(void){  int n1=5, n2=7;  n2 = producto(n1,n2);  printf ("%i, %i", n2, n1);  return 0;  }  int producto (int a,int b){  a= 20;  return (a \* b);  } |

* Considerar el siguiente fragmento de código en lenguaje de programación C. ¿Cuál o cuáles de las siguientes expresiones no es correcta?:

|  |  |
| --- | --- |
| 1. a 2. p 3. &a 4. &p 5. \*a 6. \*p 7. &\*p 8. \*&p 9. \*&a 10. &\*a | float a, \*p;  a=-4.55;  p=&a; |

* Supongamos el siguiente fragmento de código en C. Después de la ejecución de este fragmento, ¿cuál o cuáles de las siguientes afirmaciones es correcta?:

|  |  |
| --- | --- |
| 1. A vale 4 y B vale 1 2. A vale 4 y B vale 16 3. A vale 4 y B vale 1024 4. A vale 16 y B vale 1 5. A vale 16 y B vale 64 6. A vale 16 y B vale 4096 7. Error en tiempo de ejecución | void CuaCubo(int \*x, int \*x2, int \*x3){  \*x2 = \*x\*(\*x);  \*x3 = \*x\*(\*x)\*(\*x);  }  ………  B = 1;  A = 4;  CuaCubo(&A, &A, &B); |

* Dada la siguiente definición de función en C, Indicar el resultado en pantalla de la siguiente activación:

saludo(3);

|  |  |
| --- | --- |
| 1. HolaHolaHola 2. Hola   Hola  Hola   1. Hola Hola Hola 2. Error de compilación 3. Ninguna de las respuestas anteriores es correcta | void saludo(void){  printf("Hola\n");  }  …  saludo(3);  … |

* Dado el siguiente programa: ¿cuál ó cuáles de las siguientes líneas de código podría escribirse en la línea de guiones (----------- ) para que el programa compilase sin errores?

|  |  |
| --- | --- |
| 1. printf ("%f", mi\_funcion(a,b,letra)); 2. printf ("%f", mi\_funcion(&a,&b,&letra)); 3. printf ("%f", mi\_funcion(b,a,letra)); 4. printf ("%f", mi\_funcion(\*b,\*a,\*letra)); 5. El programa no compila porque la sentencia previa de asignación int num\_b=4.0; es incorrecta. | #include <stdio.h>  float mi\_funcion(float a, int b, char c);  int main(){  float a=5.5;  int b=4.0;  char letra;  -------------------  return(0);  }  float mi\_funcion(float a, int b, char c){  c='k';  return(a+b);  } |

* 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:

|  |  |
| --- | --- |
| * 1. La variable a vale 2   2. La variable a vale 3   3. La variable a vale 6   4. La variable a vale 9   5. Error de compilación: el lenguaje C prohíbe el uso de variables externas.   6. Todas las respuestas anteriores son incorrectas | #include <stdio.h>  int a;  void Func\_1(int \*b){  a=3;  \*b=a\*(\*b);  }  int main(){  a=2;  Func\_1(&a);  printf("a=%d\n",a);  return 0;  } |

* Dada la siguiente función recursiva: ¿Cuál o cuáles de las siguientes afirmaciones es correcta?

|  |  |
| --- | --- |
| 1. Al llamar a fact(0), escribe en pantalla 1 2. Al llamar a fact(1), escribe en pantalla 1 3. Al llamar a fact(2), escribe en pantalla 1 2 4. Al llamar a fact(3), escribe en pantalla 1 2 6 5. Hay un error al compilar la función 6. Todas las afirmaciones anteriores son incorrectas | int fact(int x){  if (x==0)  return (1);  else{  x = x\*fact(x-1);  printf (" %d",x);  return (x);  }  } |

* Considere el siguiente prototipo de una función C que calcula la integral definida de una función f de R en R, en el intervalo de la recta [a,b]:

double integral(double (\*f)(double x),double a,double b);

Cuál o cuáles de las siguientes instrucciones no es sintácticamente correcta:

1. printf(" Area= %.3f" ,integral(sin,0, M\_PI/2.0));
2. printf(" Area= %.3f" ,integral(sin(x),0, M\_PI/2.0));
3. printf(" Area= %.3f" ,integral(exp,0, 10));
4. printf(" Area= %.3f" ,integral(exp\*sin,0, M\_PI));
5. printf(" Area= %.3f" ,integral(&sin,0, M\_PI/2.0));
6. printf(" Area= %.3f" ,integral(\*sin,0, M\_PI/2.0));
7. Todas las anteriores son correctas

* Dado el siguiente vector de enteros inicializado con los correspondientes valores, indique ¿qué opción (u opciones) utilizaría para imprimir por pantalla el número 10?.

static int vector[10]={3, 20, 4, 5, -2, 6, 7, 10, 15, 0};

* 1. printf("%d", &(vector +8));
  2. printf("%d", \*(vector+8));
  3. printf("%d", &vector[8]);
  4. printf("%d", vector[8]);
  5. printf("%d", \*vector[8]);
  6. Ninguna de las anteriores.
* De las siguientes opciones de declaración de nuevos tipos de datos, indicar cuál o cuáles seleccionaría para poder representar en C las temperaturas (en grados centígrados y con dos decimales de precisión) del combustible que va por una tubería de 1.5 km de longitud, medidas en ambos extremos de la misma y en puntos equidistantes 50 m: tipo a;

1. typedef double tipo[1..30];
2. typedef double tipo[0..29];
3. typedef double tipo[29];
4. typedef double tipo[30];
5. typedef double tipo[31];
6. Ninguna de las anteriores

* Indique ¿cuál o cuáles de las siguientes afirmaciones respecto a un “array” en C son correctas?:
  1. Los datos de los “arrays” deben procesarse elemento a elemento empezando siempre por el primero.
  2. Un “array” completo no puede imprimirse directamente como si fuera un bloque.
  3. Un “array” no puede contener elementos que sean a su vez “arrays”.
  4. Un array puede compararse directamente con otro usando los operadores relacionales siempre que los dos contengan el mismo tipo de datos y sean del mismo tamaño.
  5. Todas las afirmaciones anteriores son incorrectas.
* ¿Qué se imprimirá al ejecutar este programa?

|  |  |
| --- | --- |
| * 1. 25 4 3 8 10 11   2. 4 3 8 10 11   3. 11 10 8 3 4 25   4. 10 8 3 4 25   5. Error de compilación   6. Error en tiempo de ejecución (“stack overflow”: desbordamiento de pila) | #include <stdio.h>  void fun(int \*x, int n){  if(n==0)  return;  else{  n--;  printf(" %d", \*(x+n));  fun(x,n);  }  }  int main(){  static int x[6]={25,4,3,8,10,11};  fun(x,6);  return 0;  } |

* ¿Qué imprime el siguiente programa?

|  |  |
| --- | --- |
| * 1. lunes martes miercoles jueves viernes sábado   2. lunes martes miercoles jueves viernes sabado domingo   3. lunes martes miercoles jueves viernes   4. viernes jueves miercoles martes lunes   5. Error de compilación   6. Error en tiempo de ejecución (“stack overflow”: desbordamiento de pila) | #include <stdio.h>  #include <string.h>  void fun(int i, char p[][10]);  int main(){  static char p[7][10]={"lunes","martes",  "miercoles","jueves","viernes",  "sabado","domingo"};  fun(0,p);  return 0;  }  void fun(int i, char p[][10]){  if(strcmp(p[i],"sabado")==0)  return;  fun(i+1,p);  printf("%s ", p[i]);  } |

* En el lenguaje C, para implementar un tipo de datos registro con parte variante se necesita:
  1. La definición de al menos un tipo “struct” y varios tipos “union” (uno para cada una de las variantes).
  2. La definición de varios tipos struct (uno para cada variante) o bien de un único tipo “union”.
  3. Sólo la definición de un tipo “struct” y un tipo “union” para representar todas las variantes.
  4. La definición de un tipo “struct”para el registro, de un tipo “union” para todas las variantes, y posiblemente varios tipos “struct” para la configuración de campos de cada una de las variantes.
  5. No es posible representar registros con parte variante en lenguaje C.
* Dado el siguiente código en lenguaje C indique ¿qué valor tomaría la variable X.B después de ejecutar el programa?:

|  |  |
| --- | --- |
| * 1. 3.4   2. 5.7   3. 5   4. 3   5. Error de compilación | #include <stdio.h>  #include <string.h>  typedef struct{  int A;  char string[10];  float B;  }ejemplo;  int main(void){  static ejemplo X={5, "string1", 3.4};  ejemplo Y;  Y.A=3;  strcpy(Y.string, "string1");  Y.B=5.7;  X=Y;  printf ("%.1f\n", X.B);  return 0;  } |

* Dada la siguiente definición de una nueva tipología de datos, ¿cómo se declararía un vector de 10 elementos de tipo ‘cancion’ y se asignan valores al primer elemento?

typedef struct{

char interprete[20];

float duracion;

}cancion;

* 1. cancion descargas[10];

strcpy (descargas.interprete[0], "U2");

descargas.duracion[0] = 20.7;

* 1. cancion descargas[10];

strcpy (descargas.interprete[0], "U2");

strcpy(descargas.duracion[0], 20.7);

* 1. typedef cancion descargas[10];

strcpy (descargas.interprete[0], "U2");

descargas[0].duracion = 20.7;

* 1. No puede realizarse, es necesario incluir valores en todos los elementos del vector.
  2. No puede realizarse, un vector no puede almacenar estructuras de datos.
  3. Todas las respuestas anteriores son incorrectas.
* ¿Cuántas iteraciones requiere el bucle del algoritmo de búsqueda binaria para localizar la posición de la primera ocurrencia del elemento de clave=15 en el vector V?

0 1 2 3 4 5 6 7 8 9 10 11 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V | 1 | 8 | 10 | 11 | 15 | 15 | 15 | 15 | 17 | 29 | 31 | 31 | 40 |

* 1. 1
  2. 2
  3. 3
  4. 4
  5. 5
  6. 6
  7. 7
  8. No se puede aplicar el algoritmo de búsqueda binaria en este vector.