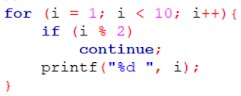
0

/

3



1

.

0

1 0

1

2

.

1

0 1

0

3

.

1

2 3

4

4

.

1

2 4

8

5

.

3 5

7

1

6

.

2

4 6

8

7

.

6

4 2

0

8

.

7

5 3

1

9

.

8

4 2

1

10

.

9

4 2

1

11

.

9

5 3

1

Respuesta correcta

8

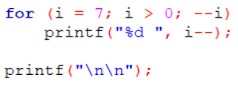
2

4 6

0

/

3



1

.

0

1 0

1

2

.

1

0 1

0

3

.

1

2 3

4

4

.

1

2 4

8

5

.

3 5

7

1

6

.

2

4 6

8

7

.

6

4 2

0

8

.

7

5 3

1

9

.

8

4 2

1

10

.

9

4 2

1

11

.

9

5 3

1

Respuesta correcta

1

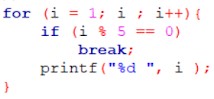
7

5 3

0

/

3



1

.

0

1 0

1

2

.

1

0 1

0

3

.

1

2 3

4

4

.

1

2 4

8

5

.

3 5

7

1

6

.

2

4 6

8

7

.

6

4 2

0

8

.

7

5 3

1

9

.

8

4 2

1

10

.

9

4 2

1

11

.

9

5 3

1

Respuesta correcta

4

1

2 3

0

/

3



1

.

0

1 0

1

2

.

1

0 1

0

3

.

1

2 3

4

4

.

1

2 4

8

5

.

3 5

7

1

6

.

2

4 6

8

7

.

6

4 2

0

8

.

7

5 3

1

9

.

8

4 2

1

10

.

9

4 2

1

11

.

9

5 3

1

Respuesta correcta

0

1

0 1

0

/

3



1

.

0

1 0

1

2

.

0 1

0

1

3

.

1

2 3

4

4

.

1

2 4

8

5

.

1

3 5

7

6

.

4 6

8

2

7

.

6

4 2

0

8

.

7

5 3

1

9

.

8

4 2

1

10

.

9

4 2

1

11

.

9

5 3

1

Respuesta correcta

9

4 2

1

Tipos de Datos Simples

de 6 puntos

0

6

/

0

El tipo de datos

𝙘𝙝𝙖𝙧

puede almacenar tanto caracteres como números

Los tipos de datos

𝙛𝙡𝙤𝙖𝙩

y

𝙙𝙤𝙪𝙗𝙡𝙚

ocupan la misma cantidad de memoria

Se puede asignar una variable de tipo

𝙛𝙡𝙤𝙖𝙩

a una variable

𝙙𝙤𝙪𝙗𝙡𝙚

sin

inconvenientes

Se puede asignar una variable de tipo de

𝙙𝙤𝙪𝙗𝙡𝙚

a una variable

𝙛𝙡𝙤𝙖𝙩

sin

inconvenientes

Es posible asignar una variable

𝙛𝙡𝙤𝙖𝙩

a una variable

𝙞𝙣𝙩

sin inconvenientes

El tipo de dato

𝙨𝙞𝙜𝙣𝙚𝙙

𝙞𝙣𝙩

ocupa lo mismo que el tipo de datos

𝙪𝙣𝙨𝙞𝙜𝙣𝙚𝙙

𝙞𝙣𝙩

Es posible asignar tanto true/false como 1/0 a una variable de tipo

𝙗𝙤𝙤𝙡

El operador

𝙨𝙞𝙯𝙚𝙤𝙛

puede utilizarse con tipos pero no con variables

Respuesta correcta

El tipo de datos

𝙘𝙝𝙖𝙧

puede almacenar tanto caracteres como números

Se puede asignar una variable de tipo

𝙛𝙡𝙤𝙖𝙩

a una variable

𝙙𝙤𝙪𝙗𝙡𝙚

sin

inconvenientes

El tipo de dato

𝙨𝙞𝙜𝙣𝙚𝙙

𝙞𝙣𝙩

ocupa lo mismo que el tipo de datos

𝙪𝙣𝙨𝙞𝙜𝙣𝙚𝙙

𝙞𝙣𝙩

Comentarios

*-*

*El tipo double ocupa el doble de espacio en memoria que oat*

*-*

*Asignar una variable de tipo double a una de tipo oat puede introducir un pérdida de*

*precisión*

*Asignar una variable de tipo oat a una de tipo int puede introducir un pérdida de*

*-*

*precisión*

*El lenguaje C no tiene un tipo de datos bool de manera prede nida*

*-*

*-*

*sizeof puede utilizarse con variables y con tipos*

Funciones

3

de 15 puntos

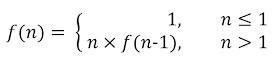
Marque aquellas afirmaciones que considere verdaderas sobre los tipos

de datos predefinidos del lenguaje C

Dada la siguiente definición matemática de función la factorial. Indicar para cada

implementación de la función si es correcta o no según las opciones de la lista

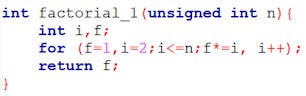
desplegable.



0

/

3



1

.

Compila y funciona correctamente

2

.

Compila, pero no funciona correctamente

3

.

Compila y genera un error en ejecución

4

.

No compila por un error en la sintaxis

5

.

Ninguna de las anteriores

Respuesta correcta

Compila y funciona correctamente

3

3

/



1

.

Compila y funciona correctamente

2

.

Compila, pero no funciona correctamente

3

.

Compila y genera un error en ejecución

4

.

No compila por un error en la sintaxis

5

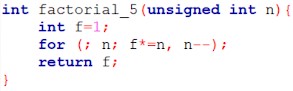
.

Ninguna de las anteriores

0

/

3



1

.

Compila y funciona correctamente

2

.

Compila, pero no funciona correctamente

3

.

Compila y genera un error en ejecución

4

.

No compila por un error en la sintaxis

5

.

Ninguna de las anteriores

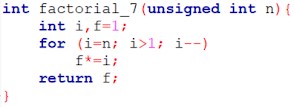
Respuesta correcta

Compila y funciona correctamente

3

0

/



1

.

Compila y funciona correctamente

2

.

Compila, pero no funciona correctamente

.

3

Compila y genera un error en ejecución

4

.

No compila por un error en la sintaxis

.

5

Ninguna de las anteriores

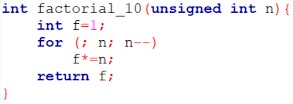
Respuesta correcta

Compila y funciona correctamente

0

/

3



1

.

Compila y funciona correctamente

2

.

Compila, pero no funciona correctamente

3

.

Compila y genera un error en ejecución

4

.

No compila por un error en la sintaxis

5

.

Ninguna de las anteriores

Respuesta correcta

Compila y funciona correctamente

Modificador Static

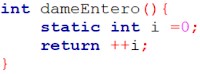
0

de 5 puntos

0

/

5



1

.

No funciona,

no puede declararse una variable static en una función

.

2

No funciona, no puede asignarse un valor a una variable static

3

.

Siempre retorna el valor 0

4

.

Siempre retorna el valor 1

5

.

Retorna la cantidad de veces que fué invocada

6

.

Ninguna de las anteriores

Respuesta correcta

Retorna la cantidad de veces que fué invocada

Punteros

3

de 18 puntos

Dada la definición de la siguiente función. Seleccione de la lista

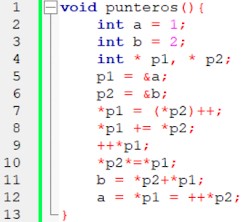
desplegable una opción que a su criterio se aproxime mejor a su

funcionamiento

Dada la implementación de la siguiente función indicar los valores que van

tomando las variables a y b según la ejecución de las instrucciones a partir de la

línea 7



3

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

10

.

a=8

b=3

11

.

a=6

b=3

12

.

a=6

b=24

13

.

a=6

b=18

\*p1 = (\*p2)++;

0

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

10

.

a=8

b=3

11

.

a=6

b=3

12

.

a=6

b=24

13

.

a=6

b=18

Respuesta correcta

a=5

b=3

\*p1 += \*p2;

0

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

.

10

a=8

b=3

11

.

a=6

b=3

12

.

a=6

b=24

13

.

a=6

b=18

Respuesta correcta

a=6

b=3

++\*p1

;

0

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

10

.

a=8

b=3

11

.

a=6

b=3

12

.

a=6

b=24

13

.

a=6

b=18

Respuesta correcta

a=6

b=18

\*p2\*=\*p1;

0

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

10

.

a=8

b=3

11

.

a=6

b=3

12

.

a=6

b=24

13

.

a=6

b=18

Respuesta correcta

a=6

b=24

b = \*p2+\*p1;

0

/

3

1

.

a=18

b=18

2

.

a=24

b=24

3

.

a=24

b=6

4

.

a=18

b=6

5

.

a=25

b=25

6

.

a=5

b=3

7

.

a=5

b=6

8

.

a=6

b=5

9

.

a=2

b=3

10

.

a=8

b=3

11

.

a=6

b=3

.

12

a=6

b=24

13

.

a=6

b=18

Respuesta correcta

a=25

b=25

Strings / Cadenas

0

de 15 puntos

Dada la siguiente implementación de la función palíndromo, que retorna 1 si una palabra se escribe igual

de derecha a izquierda como de izquierda a derecha (ej: arenera) o 0 en caso contrario. Complete las

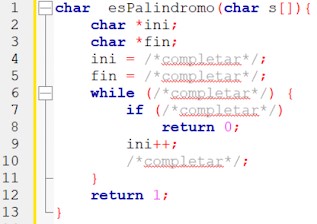
líneas de código indicadas para que la función cumpla su objetivo.

a = \*p1 = ++\*p2;

0

/

3



1

.

0

2

.

1

3

.

s

4

.

s + strlen(s)

5

.

s + strlen(s) - 1

6

.

s + strlen(s) + 1

.

7

ini

8

.

fin

9

.

fin--

10

.

fin++

11

.

ini != fin

12

.

ini == fin

13

.

ini < fin

14

.

\*ini != \*fin

15

.

\*ini == \*fin

16

.

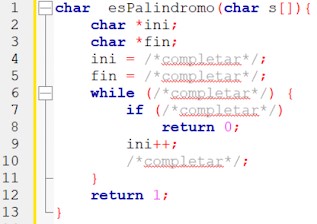
s

Línea 4: ini = ...

0

/

3



1

.

0

2

.

1

3

.

s

4

.

s + strlen(s)

5

.

s + strlen(s) - 1

6

.

s + strlen(s) + 1

7

.

ini

8

.

fin

9

.

fin--

10

.

fin++

11

.

ini != fin

12

.

ini == fin

13

.

ini < fin

14

.

\*ini != \*fin

15

.

\*ini == \*fin

16

.

\*ini < \*fin

Respuesta correcta

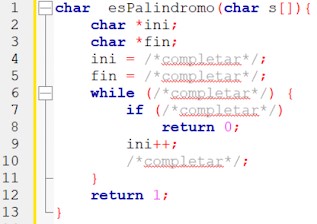
s + strlen(s) - 1

Línea 5: fin= ...

0

/

3



1

.

0

2

.

1

3

.

s

4

.

s + strlen(s)

5

.

s + strlen(s) - 1

6

.

s + strlen(s) + 1

.

7

ini

8

.

fin

9

.

fin--

10

.

fin++

11

.

ini != fin

12

.

ini == fin

13

.

ini < fin

14

.

\*ini != \*fin

15

.

\*ini == \*fin

16

.

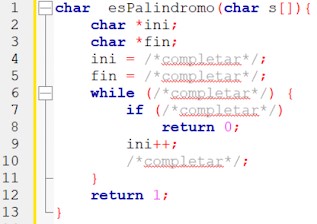
ini < fin

Línea 6: while ( ... )

0

/

3



1

.

0

2

.

1

3

.

s

4

.

s + strlen(s)

5

.

s + strlen(s) - 1

6

.

s + strlen(s) + 1

7

.

ini

8

.

fin

9

.

fin--

10

.

fin++

11

.

ini != fin

12

.

ini == fin

13

.

ini < fin

14

.

\*ini != \*fin

15

.

\*ini == \*fin

16

.

\*ini < \*fin

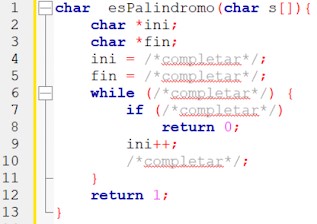
\*ini != \*fin

Línea 7: if ( ... )

0

/

3



1

.

0

2

.

1

3

.

s

4

.

s + strlen(s)

5

.

s + strlen(s) - 1

6

.

s + strlen(s) + 1

.

7

ini

8

.

fin

9

.

fin--

10

.

fin++

11

.

ini != fin

12

.

ini == fin

13

.

ini < fin

14

.

\*ini != \*fin

15

.

\*ini == \*fin

16

.

fin--

Línea 10: ...

Tipos de Datos Struct

6

de 20 puntos

Marque como verdadera o falsa cada una de las siguientes afirmaciones sobre

el tipo de datos struct.

0

/

2

.

1

Verdadera

.

2

Falsa

Respuesta correcta

Verdadera

2

2

/

1

.

Verdadera

2

.

Falsa

0

/

2

1

.

Verdadera

2

.

Falsa

Respuesta correcta

Verdadera

Es posible acceder a los campos de las estructuras en forma individual.

No es posible determinar la posición de memoria en la que se encuentra

un campo de una variable de tipo struct.

No es posible comparar variables estructura aunque sean del mismo

tipo.

2

/

2

1

.

Verdadera

2

.

Falsa

0

/

2

.

1

Verdadera

.

2

Falsa

Respuesta correcta

Falsa

2

/

0

1

.

Verdadera

2

.

Falsa

Respuesta correcta

Falsa

2

0

/

1

.

Verdadera

2

.

Falsa

Respuesta correcta

Falsa

Es posible declarar punteros a estructuras

Una estructura no puede contener otra estructura que tenga a su vez

una estructura.

Dos variables de tipo struct no pueden asignarse aunque tengan el

mismo tipo.

No puede utilizarse typedef para definir un tipo que no sea estructura.

2

0

/

.

1

Verdadera

.

2

Falsa

Respuesta correcta

Verdadera

0

2

/

1

.

Verdadera

2

.

Falsa

Respuesta correcta

Falsa

2

/

2

1

.

Verdadera

2

.

Falsa

Memoria Dinámica

0

de 6 puntos

Un arreglo de estructuras se pasa por referencia a una función de

manera automática.

Los operadores punto (.) y flecha (->) pueden utilizarse indistintamente

en una variable de tipo struct

Es posible declarar una variable de tipo struct sin nombre al momento de

definir dicha variable

0

/

6



Compila sin problemas

La ejecución de la función produce un error de memoria

La ejecución de la función introduce un riesgo potencial de violación de memoria

No compila porque tiene un error de sintaxis

La memoria reservada no podrá liberarse

La memoria reservada se inicializa con ceros

Se debe usar la función

𝙘𝙖𝙡𝙡𝙤𝙘

para reservar memoria

El parámetro

𝙥

debe estar inicializado en

𝙉𝙐𝙇𝙇

El parámetro de

𝙨𝙞𝙯𝙚𝙤𝙛

debe ser

𝙞𝙣𝙩

\*

Respuesta correcta

Compila sin problemas

La ejecución de la función introduce un riesgo potencial de violación de memoria

La memoria reservada no podrá liberarse

Comentarios

*La ejecución de la función no produce un error de memoria. Retorna de su ejecución sin*

*-*

*problemas*

*No hay errores de sintaxis en la función*

*-*

*malloc no inicializa con ceros la memoria reservada*

*-*

*podría utilizarse la función calloc para reservar memoria pero no es necesario*

*-*

*-*

*no hay necesidad de inicializar el parámetro p en NULL porque se le da valor dentro de la*

*función*

*sizeof debe retornar el tamaño del dato a reservar, en este caso un int. int \* es un puntero*

*-*

*a la memoria reservada*

Dada la implementación de la función

𝗿𝗲𝗲𝗿𝘃𝗮

que alloca memoria para

un vector de

𝗻

enteros que será usado en el programa principal. Marque

las opciones que considere correctas sobre el comportamiento de la

función.

Google no creó ni aprobó este contenido. - [Condiciones del Servicio](https://policies.google.com/terms) - [Política de Privacidad](https://policies.google.com/privacy)

[Formularios](https://www.google.com/forms/about/?utm_source=product&utm_medium=forms_logo&utm_campaign=forms)