Consignas teóricas práctica parcial

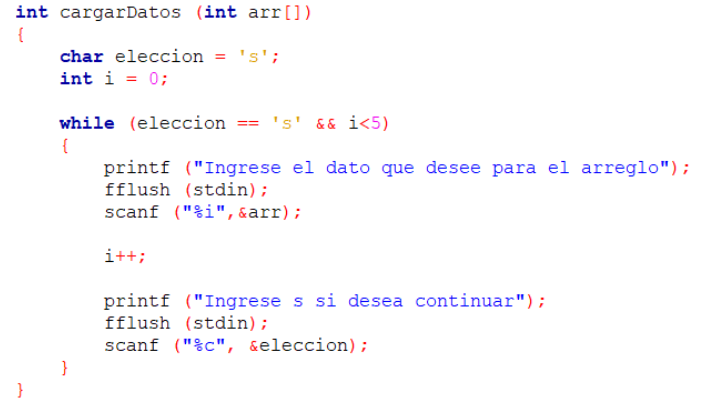
1. Responde las siguientes preguntas, justificando tanto como sea posible:
2. ¿En qué se parecen la función tope() y la función desapilar()?¿Y qué diferencias tienen?
3. ¿Qué diferencias existen entre las variables globales y las locales?
4. ¿Que podemos guardar dentro de un int,float y char? ¿Son algoritmos?
5. ¿Un puntero es una variable? ¿Que almacena?
6. ¿Que es una palabra reservada? ¿Puedo utilizarlas como identificador?

f) ¿Es posible ordenar palabras alfabéticamente? ¿Como?

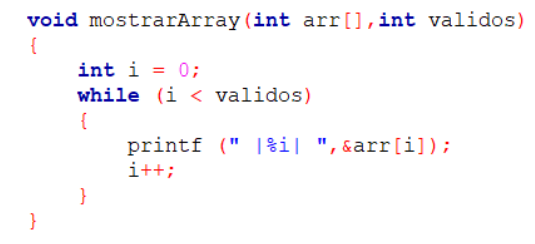
g) ¿Qué diferencias existen entre una variable, una constante y un parámetro?

h) ¿Que es un parámetro por copia y un parámetro por referencia?

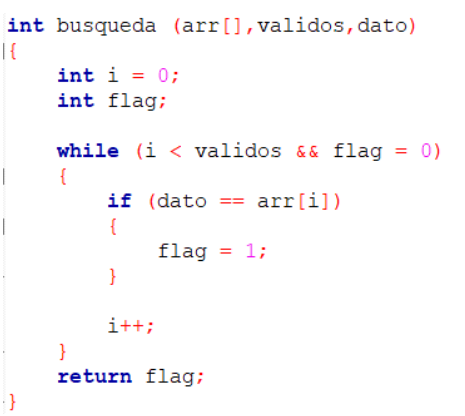
1. ¿Funciona correctamente el siguiente código?Justifique su respuesta



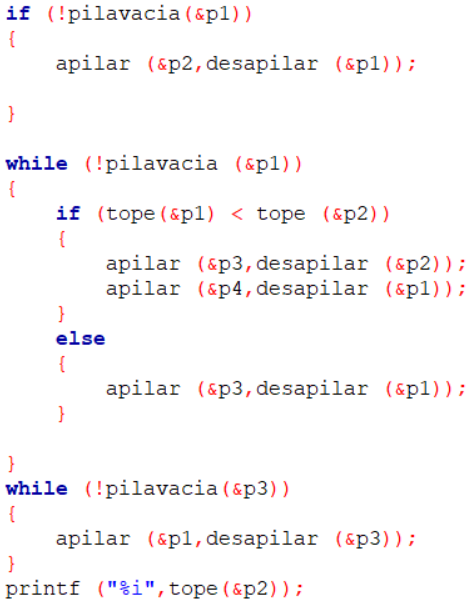
1. ¿Funciona correctamente el siguiente código?Justifique su respuesta



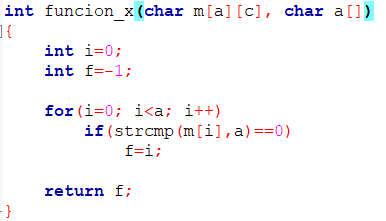
1. ¿Funciona correctamente el siguiente código?¿Hay algo que podría mejorarse? Justifique su respuesta



1. ¿Funciona correctamente el siguiente código?¿Hay algo que podría mejorarse? Justifique su respuesta



1. ¿Funciona correctamente el siguiente código?¿Que es lo que hace o pretende hacer la función?.



1. Si yo tuviese el cuerpo de la siguiente función:

int haceAlgo (int \*a, int b, int \*c)

{

\*a = b;

\*c = \*a;

return \*c;

}

En la secuencia return \*c, ¿Estoy devolviendo una dirección de memoria o solo un valor?

1. La dirección de memoria que llega por parámetro
2. Solo un valor.
3. Si yo tuviese el cuerpo de la siguiente función:

Int haceAlgo (int \*a, int b, int \*c)

{

\*a = b;

\*c = \*a;

return c;

}

En la secuencia return \*c, ¿Estoy devolviendo una dirección de memoria o solo un valor?

1. La dirección de memoria que llega por parámetro
2. Solo un valor
3. Seleccione la respuesta correcta para cada enunciado
4. El algoritmo de ordenación por inserción…

b) El algoritmo de buscar el menor…

c) El algoritmo de ordenación por selección…

Posibles respuestas:

1. Intercambia la posición del menor a la primera posición para ordenarlos.

2. Elimina un elemento.

3. Selecciona los números mas grandes y los elimina

4. Busca el menor devuelve su posición.

5. Divide un elemento por 2.

6. Busca el menor y lo imprime por pantalla.

7. Coloca el dato menor corriendo “hacia la derecha” a los demás datos.

8. Inserta números de forma aleatoria en el arreglo.

1. Seleccione la respuesta correcta para cada enunciado Cuando trabajamos con las funciones del tipo de dato Pila…
2. Apilar
3. Pilavacia
4. Desapilar
5. Tope
6. Mostrar
7. Leer

Posibles respuestas:

1. Devuelve 0 si la pila está vacía

2. Recibe el índice del valor almacenado

3. Devuelve 1 si la pila está vacía

4. Añade un valor de cualquier tipo de dato a la pila

5. Extrae el elemento que se encuentra en el tope de la pila

6. Recorre toda la pila y la imprime por consola

7. Copiar el valor del elemento que se encuentra en el tope de la pila

8. Añade un elemento pidiéndolo por teclado

9. Verifica que la pila sea distinta de 0

10. Añade un elemento de tipo entero a la pila.

1. Seleccione verdadero o falso.
2. Además de utilizar strcpy, podemos asignar valores a un arreglo de caracteres utilizando el operador =. Por ejemplo:

char arreglo [20] = “Palabra”;

char arreglo2 [20];

arreglo2 = arreglo1;

1. Para cargar por teclado un string y que este respete los espacios(ej. Ingresar “hola como estas”) debemos utilizar un gets() en vez de un scanf().
2. Solo hay una manera de declarar constantes y es como en el siguiente ejemplo: const int num=7;.
3. La unica forma de cargar un string es pidiendole al usuario por teclado.
4. La unica matriz que no debe ser obligatoria mente cargada por completo es la matriz de strings (seria un arreglo de strings).
5. El main es una función.
6. Puedes haber mas de un return en una función, pero una función no puede retornar más de una variable a la vez.
7. Todas las funciones deben tener un return.

Respuestas

1.

a. Ambas piden un dato de tipo pila y devuelven el tope, en el caso de desapilar lo saca de la pila, tope no, copia el dato y lo devuelve.

b. Las variables locales, a menos que sean pasadas por parámetro a otra función, únicamente pueden ser usada dentro de la función en las que son creadas, mientras que las globales pueden usarse en cualquier parte del código.

c. No son algoritmos, son tipos de datos primitivos, en int se guardan números enteros, float, números decimales, char, caracteres.

d. Un puntero es una variable que guarda una dirección de memoria.

e. Son palabras que se usan para el funcionamiento del código, no pueden ser usadas como identificador.

f. Si, con los algoritmos de insersion y selección.

g. Una variable es un tipo de dato que utilizas en una parte del código, una constante son variables que no pueden ser alteradas y un parámetro son las variables que piden las funciones para realizar sus tareas.

h. Cuando pasas por copia el dato no es alterado a menos que se retorne. Por referencia le pasas la dirección de memoria, quiere decir que cualquier alteración que realices afectara a la variable original.

1. En el scanf no se le pasa la posición del arreglo que queres guardar.
2. “&” en el printf, muestra la dicc. De memoria.
3. En los parámetros no de especifica el tipo de dato.
4. No, esta función busca encontrar y mostrar el menor de pila 1 sacándolo de esta en el proceso, el problema es que cuando el menor no esta en pila 2, lo desapila, haciendo que sea imposible continuar con la comparación ya que nunca se vuelve a apilar pila 2.

Para arreglarlo tenes que: dentro del if, apilar en pila 3 desapilar en pila 2, y apilar en pila 2 y desapilar en pila 3, deshaciéndote del pila 4.

1. Si. Busca una palabra en un arreglo de strings.
2. b.
3. a.
4. a->7; b->4; c->1.
5. a->10; b->3; c->5; d->7;e->6; f->8;
6. a. Falso. b. Verdadero c. Falso. d. Falso. e. Verdadero. f. Verdadero. g. Verdadero. h. Falso.