Arreglos: preguntas conceptuales y snippets para analizar I

Preguntas generales:

- ¿Cuál es la ventaja de tener datos agrupados en arreglos?
- ¿Qué propiedades comparten los datos almacenados en arreglos?
- Cite ejemplos donde la utilización de arreglos reduzca la complejidad en la codificación de un algoritmo
- ¿Cómo se define e inicializa un arreglo unidimensional?
- ¿Cómo se define e inicializa un arreglo de dos dimensiones?
- ¿Para qué se utiliza la directiva #define ?

Arreglos: preguntas conceptuales y snippets para analizar II

Considerando el siguiente snippet, analice:

```
1    float array[5]={0};
2    array[0]=10.30;
3    array[1]=10.20;
4    array[2]=-190.58;
5    array[3]=0.5;
```

- ¿Cuál el valor de inicialización de los elementos del arreglo?
- ¿Cuál es y que valor tiene el el primer elemento del arreglo?
- ¿Cuál es y que valor tiene el el último elemento del arreglo?

Arreglos: preguntas conceptuales y snippets para analizar III

Analice la siguiente porción de código:

```
1    float array[5]={0};
2    array[0]=10.30;
3    array[1.0]=10.20;
4    array[2.0]=-190.58;
```

- ¿Existen errores o el programa funciona?
- ¿Cuál es la relación entre el tipo de dato del arreglo y el valor de los índices?
- ¿Qué tipo de dato representa el valor de los índices?

Arreglos: preguntas conceptuales y snippets para analizar IV

Analice el siguiente snippet y responda:

```
1    float array[5]={0};
2    int a=2;
3    int b=1;
4    array[0]=10.30;
5    array[1]=10.20;
6    array[a+b]=-190.58;
7    array[a+b+1]=-190.58;
8    array[b-1]=0.5;
```

- ¿Cuál es el valor que tiene cada elemento del arreglo luego de la línea 8?

Arreglos: preguntas conceptuales y snippets para analizar V

Analice el siguiente snippet y responda:

```
1    int ii;
2    float array[5]={0};
3    for (ii=0;ii<10;i++)
4    {
5       array[ii]=ii;
6    }
```

- ¿Cuál es el valor que tiene cada elemento del arreglo luego de la línea 6?
- ¿Existe algún problema en el código?

Arreglos: preguntas conceptuales y snippets para analizar VI

Analice el siguiente snippet y responda:

```
1    int ii;
2    int array[5]={0};
3    for (ii=2;ii<10;i++)
4    {
5       array[ii]=ii;
6    }
```

- ¿Cuál es el valor que tiene cada elemento del arreglo luego de la línea 6?
- ¿Existe algún problema en el código?

Arreglos: preguntas conceptuales y snippets para analizar VII

Analice el siguiente snippet y responda:

```
1    int ii;
2    int array[5]={0};
3    for (ii=1;ii<4;i++)
4    {
5       array[ii]=ii*7;
6    }
```

- ¿Cuál es el valor que tiene cada elemento del arreglo luego de la línea 6?
- ¿Existe algún problema en el código?

Arreglos: preguntas conceptuales y snippets para analizar VIII

Analice el siguiente snippet y responda:

```
1  int array[5]={1,2,3,4,5};
2  float dato=0;
3  dato=(array[4]+array[3]) / (array[1+1])
```

- ¿Qué valor tiene la variable dato luego de la ejecución de la línea 3?

Arreglos: preguntas conceptuales y snippets para analizar IX

Analice el siguiente snippet y responda:

```
1  int ii;
2  int array2d[2][3]={{1,2},{3,4},{5,6}};
3  array2d[0][1]=90;
4  array2d[1][1]=190;
```

- ¿Cuántas filas y cuántas columnas tiene el arreglo declarado?
- ¿Cuál es el valor que tiene cada elemento del arreglo luego de la línea 4?

Punteros: preguntas conceptuales y snippets a analizar I

Considerando que la variable val está almacenada en la posición de memoria 0x0010, analice y responda:

```
1   int *ptr=NULL;
2   int  val=5;
3
4   ptr=&val;
5
6   printf("Valor de val %d\n",val);
7   printf("Posicion de memoria de val %p\n",&val);
8   printf("Valor de ptr %p\n",ptr);
9   printf("Posicion de memoria de ptr %p\n",&ptr);
10  printf("Val accedido por ptr %d\n",*ptr);
```

- ¿Qué imprime cada una de estas lineas?

Punteros: preguntas conceptuales y snippets a analizar II

Considerando que la variable val está almacenada en la posición de memoria 0x0010, analice y responda:

```
1   int *ptr=NULL;
2   int  val=5;
3
4   ptr=&val;
5   *ptr=*ptr + 10;
6
7   printf("Valor de val %d\n",val);
8   printf("Posicion de memoria de val %p\n",&val);
9   printf("Valor de ptr %X\n",ptr);
10  printf("Posicion de memoria de ptr %p\n",&ptr);
11  printf("Val accedido por ptr %d\n",*ptr);
```

- ¿Qué imprime cada una de estas lineas?

Punteros: preguntas conceptuales y snippets a analizar III

Considerando que la variable val está almacenada en la posición de memoria 0x0010, analice y responda:

```
1   int *ptr=NULL;
2   char val='c';
3
4   ptr=&val;
5   *ptr='a';
6
7   printf("Valor de val %d\n",val);
8   printf("Posicion de memoria de val %p\n",&val);
9   printf("Valor de ptr %p\n",ptr);
10  printf("Posicion de memoria de ptr %p\n",&ptr);
11  printf("Val accedido por ptr %d\n",*ptr);
```

- ¿Existe algún error en el código?
- ¿Qué imprime cada una de estas lineas?

Funciones: preguntas y snippets a analizar I

Preguntas generales:

- Enumere ventajas de la utilización de funciones
- ¿Qué es el prototipo de una función?. Mencione cada una de las partes y cite ejemplos
- ¿Cómo se realiza la llamada a una función que retorna datos?
- ¿Cómo se realiza la llamada a una función que no retorna datos?
- ¿Cuál es la diferencia entre paso de datos por valor y por referencia?
- Al momento de llamar a una función que recibe parámetros por valor, ¿cómo deben separarse?, ¿qué órden deben tener?
- ¿Para qué se utiliza la sentencia return?
- ¿Puede una función retornar mas de un valor?
- ¿Puede una función tener mas de una sentencia return?

Funciones: preguntas y snippets a analizar II

Considerando el siguiente snippet:

```
1  void convertFloatToInt (int , float *);
```

- ¿La función retorna datos? ¿De qué tipo?
- ¿Cuál es el nombre de la función?
- ¿La función recibe parámetros? ¿De qué forma?

Funciones: preguntas y snippets a analizar III

Considerando el siguiente snippet:

```
1  void convertFloatToInt (int* , float);
2  int main(void)
3  {
4     int    n1 =0;
5     float n1f=0;
6     convertFloatToInt(n1,n1f);
7     return(0);
8  }
```

- ¿Es correcta la llamada a la función? ¿Qué le modificaría?
- ¿Qué parámetro puede ser modificado por la función? ¿Por qué?

Funciones: preguntas y snippets a analizar IV

Considerando el siguiente snippet:

```
1  void prom (int , int , float *);
2  int main(void)
3  {
4     int   n1 =10;
5     int   n2 =40;
6     float n1f=0;
7     convertFloatToInt(n1,n2,&n1f);
8     return(0);
9  }
10
11
12
13
14 void prom (int a, int b, float *p)
15 {
16    *p = (a + b) /2.0;
17 }
```

Funciones: preguntas y snippets a analizar V

- ¿Es correcta la llamada a la función? ¿Qué le modificaría?
- ¿Qué valor tienen las variables n1, n2 y n1f luego de la ejecución de la función?
- ¿Qué valores retorna la función prom(int, int, float *)?
- ¿Qué variables puede modificar la función prom(int, int, float *)?

Funciones: preguntas y snippets a analizar VI

Considerando el siguiente snippet:

```
1  void prom (int , int , float *);
2
3  int main(void)
4  {
5     int   n1 =10;
6     int   n2 =40;
7     float n1f=0;
8     convertFloatToInt(n1,n2,n1f);
9     return(0);
10 }
11
12
13 void prom (int a, int b, float p)
14 {
15    p = (a + b) /2.0;
16 }
```

Funciones: preguntas y snippets a analizar VII

- Modifique la función para que almacene en la variable n1f el valor del promedio de las variables n1 y n2.
- Modifique la función para que almacene en la variable n1f el promedio de n1 y n2. La función debe retornar valores

Funciones: preguntas y snippets a analizar VIII

Considerando el siguiente snippet:

```
1  void funcion(void);
2
3
4
5  int main(void)
6  {
7     int var1=10;
8     funcion();
9     printf("Var 1: %d",var1);
10    return(0);
11 }
12
13 void funcion (void)
14 {
15    int var1=20;
16    printf("Var 1: %d",var1);
17 }
```

Funciones: preguntas y snippets a analizar IX

- ¿Qué esperaría que imprima este programa?
- ¿Cuál es la diferencia entre variables locales y globales?
- ¿Qué tipo de variables hay implementadas en este programa?

Funciones: preguntas y snippets a analizar X

Considerando el siguiente snippet:

```
1  int var1=20;
2
3  void funcion(void);
4
5
6
7  int main(void)
8  {
9     funcion();
10    printf("Var 1: %d",var1);
11    return(0);
12 }
13
14 void funcion (void)
15 {
16    printf("Var 1: %d",var1);
17 }
```

Funciones: preguntas y snippets a analizar XI

- ¿Qué esperaría que imprima este programa?
- ¿Cuál es la diferencia entre variables locales y globales?
- ¿Qué tipo de variables hay implementadas en este programa?

Funciones: preguntas y snippets a analizar XII

Considerando el siguiente snippet:

```
1  int modificaNumeros (int * , int * );
2  int main(void)
3  {
4     int    n1 =10;
5     int    n2 =40;
6     int    n3 =0;
7     n3=modificaNumeros(&n1,&n2);
8     return(0);
9  }
10
11 int modificaNumeros (int *a, int *b)
12 {
13    int rdo=0;
14    *a=*a+10;
15    *b=*b+10;
16    rdo=*a + *b;
17    return(rdo); }
```

Funciones: preguntas y snippets a analizar XIII

- ¿Qué esperaría que imprima este programa?
- ¿Cuál es valor de n1, n2 y n3 luego de la ejecución de la línea 7?

Arreglos, punteros y funciones: preguntas y snippets para analizar I

Preguntas generales:

- ¿Cómo se define un puntero a un arreglo?
- ¿Una función puede recibir un arreglo por valor?
- ¿Qué debe modificarse para que una función trabaje con una matriz en lugar de un arreglo de una dimensión?

Arreglos, punteros y funciones: preguntas y snippets para analizar II

Considerando el siguiente snippet:

```
1  void cargar(int []);
2  void imprimir(int []);
```

- ¿La función recibe un arreglo? ¿de cuantas dimensiones?
- En caso de recibir un arreglo, ¿de qué manera lo hace?
- ¿Qué se debería modificar para recibir arreglos de mas o menos dimensiones?