



Universidad Simón Bolívar
Departamento de Computación
y Tecnología de la Información
Computación I
CI-2125/27 Abr-Jul 2018

La verdad necesita del poder.—En sí misma, la verdad no es una potencia, digan lo que digan los retóricos racionalistas. Por el contrario, necesita ponerse de su parte a la fuerza o ponerse ella del lado de la fuerza, pues de lo contrario perecerá siempre. Es cosa demostrada hasta la saciedad.
Aurora, Federico Nietzsche.

Segundo Examen Parcial

NOMBRE _____ CARNET _____ NOTA _____

PARTE I Responda las siguientes preguntas con respuesta cortas en el espacio indicado:

1. Escriba una declaración para cada una de las siguientes variables: (2 Pts.)

- a) **table** es un arreglo de 8 enteros.
- b) **nombres** es un arreglo de 8 arreglos de 28 caracteres.
- c) **cantidades** es un arreglo de 6 enteros largos.
- d) **matriz** es un arreglo de 8 arreglos de 16 enteros.

2. Escriba una declaración para cada una de las siguientes funciones: (2 Pts.)

- a) **mcm** que halla el mínimo común múltiplo de dos enteros.
- b) **swap** que intercambia los valores de dos enteros.
- c) **invert** que invierte los primeros n elementos de un arreglo de caracteres.
- d) **posMinmax** que deja en dos variables del programa que lo invoca las posiciones de las primeras ocurrencias del mínimo y del máximo de los n primeros enteros del arreglo de enteros que recibe.

3. Responda las siguientes preguntas en el espacio proporcionado: (4 Pts.)

- a) Escriba un ciclo **while** que imprima los números del 0 al 20 en orden decreciente.
- b) Escriba un ciclo **for** que almacene en la variable *sum* la suma de los primeros n cuadros perfectos. El entero x es cuadrado perfecto si y sólo si existe un entero k tal que $x = k^2$. $1 + 4 + 9 + \dots$.
- c) Escriba un ciclo **for** que imprima los primeros n números impares positivos:
- d) Escriba un ciclo **while** equivalente al siguiente ciclo **for**: `for(i = 0, s = 0; i < n ; s += ++i);`

PARTE II De cada uno de los siguientes enunciados marque el correcto.

(5 Pts.)

1. Se llaman argumentos a:
 - las variables que se usan para invocar una función.
 - las variables que se usan después del nombre y entre paréntesis al definir una función.
 - las expresiones que se usan para invocar una función (sub-programa).
 - las variables que se definen dentro del cuerpo de una función.
2. El comando *continue* hace que:
 - se suspenda la ejecución del ciclo más interno.
 - se omitan las instrucciones que faltan por ejecutarse del bloque del ciclo más interno donde se encuentre y se continúe con el siguiente paso.
 - se continúe con la siguiente instrucción.
 - se suspenda la ejecución de todos los ciclos donde se encuentre y se continúe en la siguiente instrucción.
3. Una variable global:
 - se declara dentro del `main()` para que sea visible en todas las demás funciones.
 - puede definirse dentro de cualquier función.
 - se define fuera de cualquier función, pero se puede declarar dentro de cualquier función.
 - debe declararse antes de todas las funciones.
4. Una variable local:
 - se define fuera de cualquier bloque y fuera de cualquier función.
 - desaparece sólo cuando se termina de ejecutar el programa.
 - se define dentro de un bloque y se destruye al cerrarse el bloque más interno que la contiene.
 - no puede tener el mismo nombre que una variable global.
5. Una función en C,
 - no se puede invocar a sí misma.
 - cuando se define, debe tener un bloque aunque sea vacío.
 - puede tener el mismo nombre si varían sus parámetros.
 - no se puede declarar más de una vez.
6. Las variables locales :
 - se inicializan con valores nulos por defecto al ser definidas.
 - no se inicializan por defecto al ser definidas.
 - conservan su valor entre cada llamada de la función en la que se definen.
 - se destruyen sólo cuando se acaba la función en la que fueron definidas.
7. En lenguaje C,
 - se usa una instrucción **break**; para salir de todos los ciclos de iteración.
 - una función declarada como **void** no puede usar la instrucción **return**;
 - las variables globales son visibles a todas las funciones.
 - lo que se almacena en los parámetros de las funciones al ser invocadas es el valor de los argumentos.

8. Si f , g y h se declaran respectivamente como `void f(int);`, `int g(int, int);` y `int h(int x);`, y x, u, v son variables globales, luego una instrucción válida en el `main()` es:

- `if(!f(x)) g(u,5+v);`
- `v = g(u) + 2*f(x);`
- `g(f(x), h(2));`
- `while(!h(x)) f(u);`

9. Inmediatamente después ejecutarse el siguiente bloque:

```
{ int r = 1; for(k = 0; k < 5; k++) r *= k + 1;}
```

- la variable r tiene el valor 120.
- la variable r tiene el valor 24.
- la variable k tiene el valor 5.
- la variable k itene el valor 5 y la variable r vale 120.

10. Se llaman parámetros a las variables que:

- se usan para invocar una función.
- se usan para invocar una función (sub-programa).
- se definen dentro del cuerpo de una función.
- se usan después del nombre y entre paréntesis al definir una función.

PARTE III Escriba las respuestas de estas dos preguntas dentro de los cuadros correspondientes.

1.- Sobre la siguiente función responda de forma breve y concisa:

(3 Pts.)

```
void foo(int n){
    int k = 2;
    printf("\n%d = 1", n);
    while(n != 1){
        if(n%k == 0) {
            n = n/k;
            printf("%d", k);
            continue;
        }
        k++;
    }
}
```

1. ¿Qué hace esta función?
2. ¿Qué pasa cuando se ejecuta la instrucción `foo(20);`?
3. ¿Qué pasa si se recibe como argumento un valor de n menor o igual que cero?

2.- En sus propias palabras, defina qué es una variable global en el Lenguaje C e indique todo lo que sepa sobre ella.

(2 Pts.)

PARTE IV Escriba un diagrama de flujo y su código en C de una función de nombre `posMaxPrimo()` que recibe un arreglo de enteros, y un número entero positivo n , y devuelve la posición de la primera ocurrencia del número primo más grande en las primeras n casillas del arreglo, o -1 si no hay primos en dichas casillas. Asuma que tiene una primitiva booleana `esPrimo(k)` y una función de nombre `esPrimo(k)` que devuelve 1 si k es primo y 0 en caso contrario. Nota: en su diagrama de flujo use sub-índices, por ejemplo a_k , para referirse a la k -ésima casilla del arreglo. (7 Pts.)