

AEDD - Guía Práctica 4

Funciones - Pasaje de Parámetros por Copia

Ejercicios Propuestos

1. Escribir una función llamada linea() que reciba 2 parámetros: uno de tipo entero y otro de tipo char. La función debe mostrar una línea con tantos caracteres repetidos como indique el parámetro entero.

Ejemplos:

linea(7, '-'); debe mostrar: -----

linea(10, '#'); debe mostrar: #####

2. Escribir 3 funciones. Cada una debe recibir dos parámetros, uno de tipo entero y otro de tipo char, y deberá dibujar en pantalla la figura geométrica correspondiente según lo que se muestra a continuación:

<p>cuadrado(5, '-') :</p> <pre> - </pre>	<p>tri_sup(6, '#') :</p> <pre> # </pre> <p>Triángulo rectángulo superior izquierdo</p>	<p>tri_inf(7, '*') :</p> <pre> * </pre> <p>Triángulo rectángulo inferior izquierdo</p>
---	--	--

3. Escribir una función llamada cantDigitos que reciba un parámetro de tipo entero y retorne un entero que representa la cantidad de dígitos que contiene el parámetro.

Ejemplos:

cantDigitos (7); Retorna: 1

cantDigitos (10349); Retorna: 5

4. Escribir un programa que visualice un menú de opciones que permita elegir entre las operaciones menor, mayor, promedio y suma de 3 números enteros. Completar las funciones faltantes al código provisto por la cátedra (AEDDpr-FuncCopia-Ejercicio2.cpp).
5. Escribir una función SUMULPOT() que reciba 3 argumentos de entrada (enteros A, B y C). A y B son positivos con A<B. La función debe retornar:

(a) Si C es positivo, la sumatoria de los números en A..B, con incrementos de C.

Ejemplo:

`SUMULPOT(1, 10, 2)` debe retornar el resultado de: $1+3+5+7+9$

(b) Si C es negativo, la productoria de los números en B..A, con decrementos de -C.

Ejemplo:

`SUMULPOT(1, 18, -3)` debe retornar el resultado de: $18*15*12*9*6*3$

(c) en otro caso, la mayor de las dos potencias: A^B , B^A , invocando la función potencia del ejercicio anterior.

Ejemplo:

`SUMULPOT(2, 3, 0)` retorna 9

6. Una compañía desea transmitir datos por teléfono pero están preocupados de que sus teléfonos estén "pinchados". Todos sus datos se transmiten como enteros de cuatro dígitos. Escribir una función que cifre los datos para poderlos transmitir con mayor seguridad. Dicha función deberá recibir un entero de 4 dígitos y retornarlo cifrado como sigue:

(a) Sustituir cada dígito por $(\text{ese_digito}+7)\%10$.

(b) Luego intercambiar el primer y tercer dígitos luego el segundo y cuarto dígitos.

Ejemplo:

`cifrar(2345)` debe retornar 1290

7. Escribir una función en C que determine si un número dado entero, mayor o igual que 0, es o no palíndromo. Un número es palíndromo cuando su lectura es idéntica tanto al derecho como al revés. La función debe devolver un 1 si el número es palíndromo, y 0 en caso contrario.

Ejemplo de número palíndromo: `palindromo(21412) --> 1`

Ejemplo de número no palíndromo: `palindromo(16413) --> 0`

Ejercicios Complementarios

- A. Escribir una función que retorne el mayor de tres números. Considerar que los tres números son diferentes entre sí.

Ejemplo:

`mayorDeTres(4,27,13) -> 27`

- B. Escribir una función que reciba un número entre 1 y 12 correspondiente al de un mes y devuelva el número de días correspondiente a dicho mes. Nota: Si el número de mes introducido es menor que 1 o mayor que 12, retornar 0.

Nota: Suponer que no es un año bisiesto.

Ejemplo:

Si se ingresa el mes 5, se informará "El mes de MAYO tiene 31 días."

- C. Un número entero positivo se dice alternante si está compuesto de una secuencia de dígitos que alterna entre uno par y otro impar. Escribir una función que dado un número de n cifras, determine si es alternante.

Ejemplos: 18307 es alternante. 74120 no es alternante.

- D. Escribir una función que reciba una letra minúscula devuelva su ordinal según la tabla ascii. En caso de que el caracter ingresado no sea una letra minúscula, retornar 0.

Ejemplos:

- a. El usuario ingresa 'a' y el programa devuelve 1,
- b. El usuario ingresa 'm' y el programa devuelve 1.
- c. El usuario ingresa 'B' y el programa devuelve 0.
- d. El usuario ingresa '3' y el programa devuelve 0.

- E. Escribir una función que calcule la potencia de dos números (no usar pow), siendo su prototipo:

```
int potencia(int, int);
```

donde el primer parámetro es la base y el segundo el exponente.

Ejemplos: `potencia(2,3);` retorna 8
`potencia(0,7);` retorna 0

Funciones - Pasaje de Parámetros por Referencia

Ejercicios Propuestos

1. Escribir las funciones:

```
void masmas(int & a);  
void menosmenos(int & b);
```

que replicarán la funcionalidad de los operadores ++ y -- (incrementar y decrementar en 1 el valor de una variable entera).

Ejemplo:

```
int a=3;  
masmas(a);  
cout << a; → debe mostrar 4
```

2. Escribir la función

```
void chrtoupper( char & c);
```

que convierta un caracter en su equivalente en mayúsculas.

Ejemplos:

```
chrtoupper('a') → 'A'  
chrtoupper('B') → 'B'  
chrtoupper('1') → '1'  
chrtoupper('$') → '$'
```

3. Dadas las coordenadas X,Y de un barco, crear la función

```
void Mover(float &posX, float &posY, char direccion, float amplitud);
```

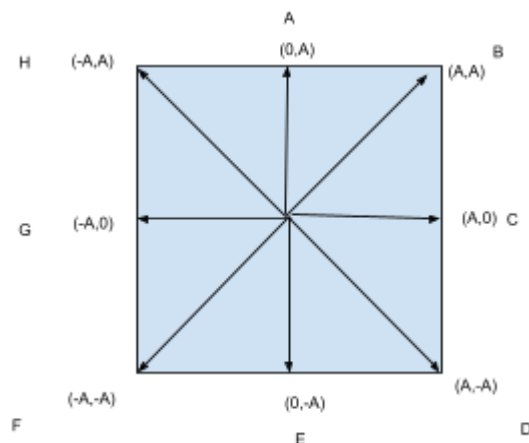
que permita actualizar la posición del mismo en base a una dirección de movimiento (un caracter entre la 'A' y la 'H', ver imagen) y la amplitud (A) del cambio de posición.

Ejemplo:

```
posX = 1.5;
```

```
posY = -5.2;
```

```
Mover(posX, posY, 'D', 1.3); → posX = 2.8 / posY = -6.5
```



4. Escribir la función

```
void diaSiguiente(int &dia, int &mes, int &anio);
```

que incremente la fecha en un día. Tener en cuenta años bisiestos y que se ingrese una fecha válida, en este último caso debe retornar ceros.

Ejemplos de invocación:

```
diaSiguiente( 1, 12, 2015) → 2,12,2015
```

```
diaSiguiente(28, 2, 2012) → 29,2,2012 (2012 año bisiesto)
```

```
diaSiguiente(28, 2, 2014) → 1,3,2014
```

```
diaSiguiente(28, 22, 2014) → 0,0,0
```

5. Diseñar una función que reciba como parámetros tres números reales: a, b y c. La función debe indicar el menor y el mayor, retornando además un código que indique 1, 0 o -1 según si todos los números son distintos, si hay un par de números iguales ó si los tres números son iguales, respectivamente. Si todos los números son iguales, no existe menor ni mayor.

Ejemplos de invocación:

```
accion( 5.8, 4.7, 1.0, menor, mayor) -> Retorna 1 y coloca los valores de menor y mayor de la siguiente manera: menor = 1.0, mayor = 5.8.
```

```
accion(-7.0, -7.8, -7.0, menor, mayor) -> Retorna 0 y coloca los valores de menor y mayor de la siguiente manera: menor = -7.8, mayor = -7.0.
```

```
accion( 4.7, 4.7, 4.7, menor, mayor) -> Retorna -1.
```

6. Escribir una función llamada division que reciba como parámetros dos numeros reales dividendo y divisor y devuelva, si la división es factible, en un parámetro de salida el resultado de ejecutar la división entre dividendo y divisor. Además, la función debe retornar verdadero si la división pudo

realizarse con éxito (falso en caso contrario).

Funciones Recursivas

Ejercicios Propuestos

1. Escribir una función recursiva que calcule y retorne la cantidad de dígitos de un número entero.

Ejemplos:

`cantDigitos(1234) → 4` `cantDigitos(0) → 1`

Variantes:

- Escribir una función recursiva que calcule la cantidad de dígitos pares de un número entero.

Ejemplos:

`cantDigPares(1234) → 2` `cantDigPares(44) → 2`
`cantDigPares(1) → 0` `cantDigPares(4) → 1`

- Escribir una función recursiva que indique si un número entero tiene algún dígito par.

Ejemplos:

`tieneDigPares(1234) → true` `tieneDigPares(44) → true`
`tieneDigPares(1) → false` `tieneDigPares(4) → true`

- Escribir una función recursiva que calcule la suma de los dígitos de un número entero que son múltiplos de N.

Ejemplos:

`sumaDigMul(1234,2) → 6` `sumaDigMul(44,4) → 8`
`sumaDigMul(7,3) → 0` `sumaDigMul(4,4) → 4`

2. Escribir una función recursiva que devuelva la suma de los primeros N enteros.

Ejemplos:

`suma(5) → 1+2+3+4+5 = 15`
`suma(1) → 1`

Variantes:

- Escribir un programa que retorne la suma de los enteros positivos pares desde N hasta 2.

Ejemplos:

`sumaPares(12) → 12+10+8+6+4+2= 42` `sumaPares(11) → 10+8+6+4+2= 30`
`sumaPares(1) → 0` `sumaPares(3) → 2`

3. Dado un número entero largo, imprimirlo en orden inverso. Resolverlo a través de una función recursiva.

Ejemplos:

`inverso(35891) → 1 9 8 5 3`
`inverso(7) → 7`

4. Escribir una función recursiva que nos diga si un número es capicúa.

Ejemplos:

`capicua(35891) → false`

`capicua(35853) → true`

`capicua(7) → true`

5. Escribir una función que sea recursiva, con nombre `maxDigito(n)`, que reciba un número natural largo y devuelve el dígito más grande del número.

Ejemplos:

`maxDigito(35897) → 9`

`maxDigito(7) → 7`

6. Determinar el producto de dos números a través de la suma y de manera recursiva.

7. Escribir una función recursiva que calcule la potencia de un número.

Ejemplos:

`potencia(2,5) → 32`

`potencia(71,1) → 71`

`potencia(13,0) → 1`

8. Escribir una función recursiva que transforme un número entero positivo a notación binaria.

Ejemplos:

`decBinario(41) → 101001`

9. Escribir una función recursiva que transforme un número expresado en notación binaria a un número entero.

Ejemplos:

`BinDec(101001) → 41`

10. Un granjero ha comprado una pareja de conejos para criarlos y luego venderlos. Si la pareja de conejos produce una nueva pareja cada mes y la nueva pareja tarda un mes más en ser también productiva, ¿cuántos pares de conejos podrá poner a la venta el granjero al cabo de un año?

11. Las secuencias de `ccanobif` son secuencias en las que cada término es siempre igual a la suma de los dos siguientes. Excepto por los dos últimos términos que siempre son iguales a 1. Dado un valor entero, imprimir la secuencia `lccanobif` de tamaño correspondiente.

Ejemplos:

`ccanobif(10) → 55 34 21 13 8 5 3 2 1 1`

`ccanobif(5) → 5 3 2 1 1`

12. Implementar una función recursiva `cuentaRegresiva(sequencia)` que informe una secuencia de despegue de una nave espacial, con el siguiente comportamiento:

- Si los segundos que faltan para el despegue (el único parámetro de la función) son menos que 10, se visualiza ese valor, se espera un segundo y se continúa. Si falta 0 segundos, se visualiza "DESPEGUE" y se termina.
- Si faltan más de 10 segundos, sólo hay que informar la cantidad de segundos que faltan para el despegue, cada 10 segundos, cuando el dígito unidad es 0 (y hay que esperar la cantidad de segundos que corresponda).

Para la espera utilizar la función Sleep(mseg) de la librería <windows.h>

Ejemplos:

```
cuentaRegresiva(35) --> 30 - 20 - 10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 -  
DESPEGUE  
cuentaRegresiva(6) --> 6 - 5 - 4 - 3 - 2 - 1 - DESPEGUE
```

13. Dados dos números naturales n y b con $b < n$, podemos hacer que n explote usando a b como bomba. Cuando n explota se parte en dos números naturales $n1 = (n \text{ div } b)$ y $n2 = (n - n1)$ y se produce una reacción en cadena. Si $n1$ (o $n2$) es mayor que b , también explota y se parte nuevamente en dos pedazos, siguiendo el criterio anterior. Esto se repite hasta que todos los pedazos resultantes de la explosión de n sean menores o iguales que b . Diseña y escribe una función recursiva que, dados dos números n y b , muestre todos los pedazos de la explosión de n usando a b como bomba.

Ejemplos:

Si $n=23$ y $b=3$, los pedazos son 2 1 1 3 1 1 3 3 2 2 1 3 puesto que las explosiones generan: (7,16), luego cada uno ((2,5), (5,9)) y así sucesivamente.