

AEDD- Guía Práctica 6: Funciones - Pasaje de parámetros por referencia

Se solicita codificar una función en C++ para cada uno de los siguientes ejercicios. En todos los casos considerar los parámetros de las funciones y sus valores de retorno.

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 de uso:

```
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.

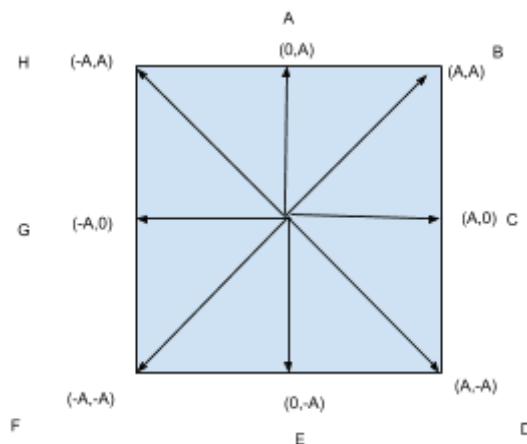
Ejemplos:

```
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= 1.5 + 1.3 = 2.8
                                posY= -5.2 - 1.3 = -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.
```