

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

TALLER DE ARREGLOS N-DIMENSIONALES FUNDAMENTOS DE PROGRAMACIÓN

Tema 1.- Segundo valor mínimo y máximo

Escriba un programa en python que reciba tres números por teclado. El primer número va a representar el tamaño del arreglo, los dos siguientes números representan el rango de los números aleatorios que conformarán el arreglo. Finalmente, usted deberá presentar por pantalla el segundo número mínimo y el segundo número máximo. A continuación, se muestra un ejemplo de la salida del programa.

```
Ingrese el tamaño del arreglo: 5
Ingrese el mínimo del rango de los números aleatorios: 1
Ingrese el máximo del rango de los números aleatorios: 20
[ 3  2 11 11  8]
El segundo valor mínimo es:  3
El segundo valor máximo es:  8
```

Tema 2.- Cambiando esquinas

Escribir un programa en python que reciba por teclado un número de filas y un número de columnas por teclado y a partir de esos números cree una matriz aleatoria con números aleatorios entre 10 y 20. Luego, solicite un número y utilice este número para remplazar las esquinas de la matriz por el valor que ingresó el usuario.

```
Ingrese el número de filas de la matriz: 3
Ingrese el número de columnas de la matriz: 5
Ingrese el valor que tendrá las esquinas: 100
[[14 14 17 19 14]
 [10 18 10 17 16]
 [18 20 18 16 17]]
Matriz cambiada las esquinas:
[[100 14 17 19 100]
 [ 10 18 10 17 16]
 [100 20 18 16 100]]
```

Tema 3.- Cambiando bordes

Escribir un programa en python que reciba por teclado un número de filas y un número de columnas por teclado y a partir de esos números cree una matriz aleatoria con números aleatorios entre 5 y 25. Luego, solicite un número y utilice este número para reemplazar los bordes de la matriz por el valor que ingresó el usuario.

```

Ingrese el número de filas de la matriz: 3
Ingrese el número de columnas de la matriz: 5
Ingrese el valor que tendrá las esquinas: 100
[[16  9 23 11  5]
 [24 24 15 24  7]
 [ 8 17 10 15 19]]
Matriz cambiada las bordes:
[[100 100 100 100 100]
 [100  24  15  24 100]
 [100 100 100 100 100]]

```

Tema 4.- Battleship

Escriba un programa en python que simule una variante del juego del battleship (batalla de barcos). Usted debe solicitar por teclado el número de barcos que se colocarán en el tablero, además, debe solicitar la dimensión del tablero. La dimensión del tablero está dada por filas y columnas, estos valores deben ser mayor al número de barcos, en caso de que el usuario no cumpla con esta condición, debe solicitar el reingreso de las filas y columnas.

Los barcos ocupan 3 posiciones horizontales en el tablero y deben ser colocados en el tablero de forma aleatoria, cada barco tiene un identificador que comienza en 1 hasta el número de barcos que ingreso el usuario. A continuación, se muestra un ejemplo de lo citado (*El valor de cero representa el mar*):

```

Ingrese la cantidad de barcos: 5
Ingrese el número de filas del tablero: 3
Ingrese el número de columnas del tablero: 6
!!!!Error las dimensiones del tablero debe ser mayor al número de barcos
Ingrese el número de filas de la matriz: 6
Ingrese el número de columnas de la matriz: 7
[[0 0 0 0 0 0 0]
 [0 0 0 0 3 3 3]
 [0 0 5 5 5 0 0]
 [0 0 0 0 0 0 0]
 [0 4 4 4 0 0 0]
 [2 2 2 0 1 1 1]]

```

Para la simulación del juego, el usuario cuenta con una cantidad de disparos inicial que es calculado a partir de la mitad del número de barcos que ingresó el usuario.

Para disparar debe solicitar la posición bajo el formato **3,4** (Esto representa fila 3 y columna 4).

La munición siempre es una bomba de expansión a la redonda, es decir, si el usuario dispara a la posición 3,4, esto afectará a lo que se encuentre en la posición 2,4 – 4,4 – 3,2 – 3,3 – 3,4 – 5,2 – 5,3 – 5,4.

Cuando un barco sufrió un disparo este se hunde, es decir, solo debería verse el mar.

El juego termina cuando se hayan eliminado todos los barcos o cuando el usuario ya no tenga municiones.

Al final se debe presentar por pantalla cuantos barcos eliminó el usuario. También debe presentar el puntaje final obtenido por el usuario ($\text{barcos eliminados} * 100 + \text{tiros restantes} * 200$).