



UNIVERSIDAD
COMPLUTENSE
MADRID

Fundamentos de la Programación I

Ayuda de la Práctica (Versión 1)

(Basado en la práctica de Luis Garmendia, Ramón González del Campo, Pablo Rabanal y Luis Hernández)

Índice

1. Introducción
2. Planificación
3. Plantilla de la Solución y Main
4. Funciones aleat, toStr, fichaToStr y mostrarTablero
5. Funciones puedePoner y ponerFicha
6. Pruebas Finales

1. Introducción

- ✓ Con esta presentación se pretende ayudar en la implementación de la versión 1 de la práctica 1.
- ✓ Necesario: un mínimo de práctica con **bucles, funciones y ficheros**.
- ✓ ¡Practica con los ejercicios de los temas 2 y 3 si no lo has hecho ya!
- ✓ Esta ayuda es opcional. No es obligatorio seguirla para implementar la solución de la versión 1.

2. Planificación

- ✓ Para realizar la versión 1, debes seguir la siguiente planificación de laboratorios:
 - 23/10: Plantilla de la Solución y Main
 - 30/10: Funciones aleat, toStr, fichaToStr y mostrarTablero
 - 6/11: Funciones puedePoner, ponerFicha y Pruebas

3. Plantilla de la Solución y Main

- ✓ Prepara una **plantilla** con los prototipos de las **nueve funciones** pedidas en el enunciado y la función *main*.
- ✓ Añade las implementaciones de las funciones (de momento sin código) después de la función *main*.
- ✓ Implementa el bucle principal de *main* y la función *mostrarMenu* según la teoría vista en clase (ejemplo de switch en la transparencia 310). El bucle principal se ejecuta hasta que la opción introducida es 0.
- ✓ Prueba el bucle principal con el menú. De momento las funciones no hacen nada.

4. Funciones aleat, toStr, fichaToStr y mostrarTablero

- ✓ Introduce la llamada a la función *srand(time(NULL))* al comienzo de la función *main* (después de la declaración de variables). Cuidado con las librerías necesarias.
- ✓ Implementa la función *aleat()* usando la función *rand()* con el *limiteInferior* 0 y *limiteSuperior* 6.
- ✓ Implementa la función *toStr* teniendo en cuenta que el carácter de un número *n* es:

```
char(int('0') + n)
```

- ✓ Implementa la función *fichaToStr* concatenando barras verticales ‘|’ y ‘-’ a los resultados de llamar a la función *toStr* con los valores de los números que se pasan como parámetros.

- ✓ Implementa la función *mostrarTablero* según el siguiente pseudocódigo:

```
Mostrar ----- ;  
Mostrar | TABLERO | ;  
Mostrar ----- ;  
Mostrar el tablero  
Mostrar Fichas colocadas: numColocadas  
Mostrar Fichas robadas: numRobadas  
Mostrar Ficha jugador: fichaToStr(fichaN1, fichaN2)
```

- ✓ Introducir lo siguiente en la función *main*:
 - **variables** apropiadas para toda la ejecución.
 - Inicialización del **tablero** (antes del bucle principal).
 - Generación de la **ficha** del jugador (antes del bucle principal, al final del bucle y cuando se roba ficha).
 - Llamadas a la función *mostrarTablero* (antes y al final del bucle principal).
- ✓ Ten en cuenta que, al final del bucle, sólo se genera la ficha del jugador si no se ha robado. Utiliza una variable boolean para indicarlo.
- ✓ Probar vuestro código. Ver que se muestra el tablero con las fichas generadas (la inicial y la del jugador).

5. Funciones *puedePoner* y *ponerFicha*

- ✓ La función *puedePonerIzq* debe devolver una variable *bool* teniendo en cuenta el siguiente valor:

```
int(tablero[1]) - int('0')
```

- ✓ La función *puedePonerDer* debe devolver una variable *bool* teniendo en cuenta el siguiente valor:

```
int(tablero[tablero.size() - 2]) - int('0')
```

- ✓ Para implementar las funciones *ponerFichaIzq* y *ponerFichaDer*, concatenar al *tablero* los strings correspondientes teniendo en cuenta que puede ser necesario intercambiar *fichaN1* y *fichaN2* antes de concatenar.

6. Pruebas Finales

- ✓ Incorpora las llamadas a las funciones *puedePonerIzq*, *puedePonerDer*, *ponerFichaIzq* y *ponerFichaDer* en los casos correspondientes del switch de la función *main*.
- ✓ Ten en cuenta que sólo se “pone” ficha si es posible y que después hay que actualizar el contador de fichas colocadas.
- ✓ Si no se puede poner ficha hay que mostrar un mensaje de error.
- ✓ Prueba exhaustivamente tu código poniendo fichas a la derecha, a la izquierda, robando y saliendo del programa.
- ✓ Recuerda que la prueba funcional es muy importante para la nota final de la práctica.