Fundamentos de la Programación I

Ayuda de la Práctica (Versión 2)

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

Índice

10. Implementación de la Función PuedeColocarFicha
11. Implementación de la Función SumaPuntos
12. Modificaciones en el Switch (II)

1. Introducción

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

2. Planificación

- ✓ Para realizar la versión 2, debes seguir la siguiente planificación de laboratorios:
  - 13/11: Proyecto, Modificaciones, GeneraPozo y RobaFicha
  - 20/11: MostrarTablero, EliminaFicha y Modificaciones Switch(I)
  - 27/11: PuedeColocarFicha, SumaPuntos, Modificaciones Switch(II) y Pruebas

3. Proyecto y Modificaciones

- ✓ Crea un **nuevo proyecto** en Visual Studio para la versión 2.
- ✓ Copia el archivo *main.cpp* de la versión 1 en el directorio del nuevo proyecto de Visual Studio.
- ✓ Añade tu archivo *main.cpp* al proyecto desde el explorador de la solución (en la carpeta de fuentes).
- ✓ Pon un comentario en el archivo *main.cpp* indicando que se trata de la solución de la versión 2 de la práctica.
- ✓ No te olvides de añadir el comentario con el nombre y apellidos de los integrantes del grupo de laboratorios.

- Modifica el programa *main.cpp* añadiendo los prototipos de las nuevas funciones y la modificación de *mostrarTablero*
  - `void generaPozo(tArray pozo1, tArray pozo2, int maxValor);`
  - `void mostrarTablero(string tablero, short int numColocadas, short int numRobadas, const tArray fichas1, const tArray fichas2, short int fichasCont);`
  - `void robaFicha(const tArray pozo1, const tArray pozo2, short int &cont, short int &fichaN1, short int &fichaN2);`
  - `void eliminaFicha (tArray fichas1, tArray fichas2, short int &fichasCont, short int fichaNum);`
  - `bool puedeColocarFicha(const tArray fichas1, const tArray fichas2, short int fichasCont, string tablero);`
  - `short int sumaPuntos(const tArray fichas1, const tArray fichas2, short int fichasCont);`
- Prepara las funciones (sin código) después de la función *main*.
- En esta versión, es necesario modificar las funciones *main* y *mostrarTablero* tal y como indicamos más adelante.

- ✓ Define el tipo array *tArray* con tamaño 28.

```
const int NumFichas = 28;
typedef short int tArray[NumFichas];
```

- ✓ La constante y la definición del tipo *tArray* deben aparecer antes de los prototipos de las funciones.
- ✓ Declara variables *tArray* en la función *main()*: *pozo1*, *pozo2*, *fichas1* y *fichas2*. Declara también contadores para el pozo y las fichas del jugador: *pozoCont* y *fichasCont*.
- ✓ Modificaciones iniciales en la función *main()*:
  - Declara una variable *boolean* **salir**, inicialízala a false y utilízala en el bucle principal. En esta versión 2 no sólo se sale del bucle cuando la opción introducida por el usuario es 0.
  - Elimina las llamadas a *mostrarMenu* y *mostrarTablero* de antes del bucle principal y del final de dicho bucle. Llámalas una única vez al comienzo del bucle principal.

4. Implementación de la Función GeneraPozo

- ✓ Esta función genera el pozo de 28 fichas.
- ✓ Declara una variable *maxValor* en la función *main* y dale valor 6.
- ✓ Observa las 28 fichas (en la presentación de la versión 1) y utiliza dos bucles **for anidados** para generarlas.
- ✓ Una vez implementada la función *generaPozo*, añade su llamada en la función *main*. Ten en cuenta que el pozo se genera en el inicio del juego (antes del bucle principal).

5. Función DesordenaPozo

- ✓ Esta función está implementada en el enunciado de la práctica.
- ✓ Copia el código de la función en *main.cpp*.
- ✓ Añade la llamada a *desordenaPozo* en la función *main*(). Ten en cuenta que el pozo se desordena una vez generado. Después, inicializa los contadores de fichas del pozo y del jugador (*pozoCont* y *fichasCont*).

6. Implementación de la Función RobaFicha

- ✓ La función *robaFicha* permite coger una ficha del pozo modificando los parámetros por referencia *fichaN1*, *fichaN2* y *cont*.
- ✓ El parámetro *cont* de la función es el contador del pozo.
- ✓ Implementa la función teniendo en cuenta que se cogen los números de las últimas posiciones de los arrays y después se decrementa el contador en 1.
- ✓ Una vez implementada la función, añade su llamada en la función *main*(). Se roba una ficha del pozo para generar el primer tablero.
- ✓ Genera el primer tablero después de robar la ficha.
- ✓ A continuación roba las **7 fichas del jugador** del pozo. Utiliza un bucle **for**.

7. Modificación de la función MostrarTablero

- ✓ La nueva función *mostrarTablero* tiene la siguiente cabecera:

```
void mostrarTablero(string tablero, short int numColocadas, short int numRobadas, const tArray fichas1, const tArray fichas2, short int fichasCont)
```

- ✓ Modifica el prototipo, si no lo has hecho ya, y la implementación de la función *mostrarTablero*. Ten en cuenta que bajo los números de fichas colocadas y robadas deben aparecer las fichas del jugador. Fíjate en el ejemplo de ejecución del enunciado.
- ✓ Prueba tu código y comprueba que se ven las 7 fichas que roba el jugador.

8. Implementación de la función EliminaFicha

- ✓ Para implementar esta función, fíjate en la transparencia 556 del tema 5 que hemos visto en clase. La posición que consideramos para eliminar es *fichaNum-1*.
- ✓ La función *eliminaFicha* recorre los arrays *fichas1* y *fichas2* desde la posición *fichaNum-1* hasta *fichasCont-2* haciendo que los elementos de la posición *i* tomen el valor de los elementos de la posición *i+1*.
- ✓ Después de este bucle for, disminuye *fichasCont* en 1.

9. Modificaciones en el Switch (I)

- ✓ Modifica el switch de *main*() para solicitar un numero de ficha de jugador y coger dicha ficha antes de comprobar si se puede colocar por la izquierda o por la derecha.
- ✓ Para solicitar el número de ficha:

```
cout << "Numero de ficha [1.." << fichasCont << "]: ";
cin >> fichaNum;
```

- ✓ Comprueba que el número de ficha es correcto
- ✓ Si el número de ficha es correcto, coge *fichaN1* y *fichaN2* de la posición *fichaNum-1* de los arrays *fichas1* y *fichas2*.
- ✓ Después de poner la ficha a la izquierda o a la derecha, elimina la ficha de los arrays *fichas1* y *fichas2* llamando a la función *eliminaFicha*.

10. Implementación de la función PuedeColocarFicha

- ✓ Esta función realiza una búsqueda en *fichas1* y *fichas2* utilizando una variable *boolean* puede.

```
mientras (!puede && (i < fichasCont))
    si puede poner ficha por la izquierda o puede poner ficha por la derecha
        puede = true
    sino
        i = i + 1
```

11. Implementación de la función SumaPuntos

- ✓ Utiliza un bucle for para sumar todos los números de *fichas1* y *fichas2*.

12. Modificaciones en el Switch (II)

- ✓ Necesitáis modificar la implementación de la opción robar (case 3 del switch) teniendo en cuenta lo siguiente:

```
Si puede colocar ficha
    Mostrar “Puedes colocar fichas”
Sino si (pozoCont == 0)
    Mostrar “No hay mas fichas en el pozo”
    salir = true
Sino
   Robar ficha del pozo
   Añadir la ficha a las fichas del jugador
   fichasCont = fichasCont + 1
   numRobadas = numRobadas + 1
```

- ✓ Fuera del switch (y dentro del bucle principal), necesitáis hacer las siguientes comprobaciones

```
Si (fichasCont == 0)
     Mostrar “Has ganado!”
     salir = true
Si (pozoCont == 0) && no puede colocar ficha
     Mostrar “No se puede robar más”
     Mostrar “Los puntos totales son: “
     Mostrar sumaPuntos
     salir = true
```