



Deal or no deal es un formato de concursos creado por la productora holandesa *Endemol*.

El programa consiste en un número de maletines que representan diversas cantidades de dinero. Sin saber a que cantidad corresponde cada maletín, el concursante escoge uno, el cual se supone contiene el premio máximo.

Luego el participante va abriendo maletines uno por uno para saber el valor que representaban. Mientras ocurre eso el concursante puede recibir ofertas de la "*banca*", la cual desea comprar su maletín por el menor valor posible, el cual depende de las cantidades que queden. El concursante puede aceptar la oferta o seguir abriendo maletines. Como ambos (el concursante y la banca) desconocen el valor de su maletín, puede que el concursante haga muy buen negocio al vender a un buen precio un maletín que puede contener cifras mínimas. Si opta por seguir abriendo maletines y estos corresponden a cifras menores, aumentará el valor de la oferta de la banca.

Hay solo dos rutas para que el concursante gane el premio mayor:

1. Elegir al comienzo el maletín correcto y nunca aceptar una oferta de la banca.
2. Mantener la opción del premio máximo hasta que queden solo 2 maletines y cambiarlo por el suyo.

Se desea implementar el juego *Deal or no deal* en C++ cumpliendo con, al menos, las siguientes especificaciones:

- ✓ Constará de nueve maletines numerados del 1 al 9 con las siguientes cantidades asociadas:

- \$100
- \$500
- \$1.000
- \$2.500
- \$5.000
- \$10.000
- \$25.000
- \$50.000
- \$100.000

Los datos sobre qué cantidad de dinero estará asociada a cada maletín se encontrarán en un archivo de texto situado en el directorio del ejecutable y rotulado como *maletines_oficiales.txt* con el formato “*maletín, dinero*”. Ejemplo:

```
1, 5000
2, 100
3, 25000
4, 100000
...
```

- ✓ En la pantalla inicial del juego se mostrará el nombre del participante que más dinero ganó hasta el momento (siempre y cuando no sea la primera ejecución del programa).
- ✓ A continuación el usuario ingresa su nombre y se le mostrarán todos los maletines disponibles en formato de matriz de 3x3 para que escoja uno de ellos ingresando el número asociado al maletín.

1	2	3
4	5	6
7	8	9

A partir de aquí comienza el juego propiamente dicho:

- ✓ El usuario procederá a abrir tres maletines para conocer su valor.
- ✓ Acto seguido se recibirá la *oferta de la banca* (siempre número entero). Se supone que la banca desconoce el contenido de los maletines y debe mantenerse imparcial. El algoritmo utilizado para determinar la cuantía de la oferta quedará a cargo de la implementación y de los supuestos del programador.
- ✓ Si el usuario acepta la oferta propuesta, se procederá a mostrarle el contenido de su maletín y se guardará el registro de la plata que ha ganado en un archivo rotulado como *jugadores.txt* debajo del último participante con el formato "*jugador premio*". Ejemplo:

```
Uriel 25000
Jennifer 1350
Fortuna 100000
...
```

Considerar que en caso de ser el primer participante en jugar, el archivo debe crearse en el directorio del ejecutable.

- ✓ Si, por el contrario, el usuario rechaza la oferta y decide seguir jugando, se le deberá mostrar los maletines que quedan por abrir conjuntamente con los posibles premios que aún faltan salir y los que ya salieron. Ejemplo, si ya eligió los maletines 3, 5 y 9 cuyas cantidades asociadas eran \$2.500, \$100 y \$50.000:

1	2	
4		6
7	8	

\$100	\$500	\$1.000
\$2.500	\$5.000	\$10.000
\$25.000	\$50.000	\$100.000

- ✓ El juego continuará con la misma lógica, es decir, el jugador seleccionará ahora dos maletines de los seis posibles, se le mostrará su contenido y recibirá luego la oferta de la banca. Si decide continuar, a partir de aquí irá eligiendo de a un maletín por vez hasta decidir aceptar la oferta de la banca o llegar a la última instancia dónde sólo queda un maletín por abrir y deberá decidir si quedarse con su maletín o cambiarlo.
- ✓ Al finalizar el juego, no olvide guardar los datos en el archivo correspondiente antes mencionado.
- ✓ El juego podrá reiniciarse eligiendo otro nombre de jugador.
- ✓ Se deben realizar todas las validaciones necesarias, evitar la ejecución de ciclos innecesarios, lograr un código que sea autodocumentado con nombres de variables representativos y crear una interfaz amigable, clara, amena e intuitiva para el usuario.

Conjuntamente con el código del programa, deberá hacer entrega de un informe que explicita la estrategia de resolución llevada a cabo y aclare todos los supuestos bajo los cuales trabaja.

El trabajo es individual y la entrega del mismo será el día **miércoles 24 de agosto del 2016** vía e-mail adjuntando tanto los archivos fuentes como el informe en un .zip rotulado como **"TP1_APELLIDO_NOMBRE"**.

No se descarta la posibilidad de que el alumno tengan que hacer una defensa del trabajo presentado en caso de ser necesario.

En caso de detectarse plagio en el código, el trabajo será desaprobado sin oportunidad de recuperarlo.

Sea creativo y original.

Aclaración importante: los trabajos que no cumplan los requisitos indicados no serán aceptados.