

**Deal or No Deal – With a Twist**

**Program Description**

This assignment requires you to write a program to play a modified version of the popular game “Deal or No Deal”. Usually, the game begins with 26 unopened cases where each case corresponds to a certain amount of money. However, your program should give the player the choice of playing the game with 16, 18, 20, 22, 24, or 26 unopened cases.

A game is played for a certain number of rounds. The number of rounds depends on the number of cases chosen by the player. In the first and subsequent rounds, the player must open a certain number of cases before getting an offer from the bank. At this point, the player can choose to accept the offer and stop playing (“Deal”) or keep on playing (“No Deal”).



The monetary amounts in a game and the number of rounds in the game depend on the number of cases chosen at the beginning by the player. This information must be read from a file as follows:

# Cases in Game	Name of File Containing Relevant Data
16	Data16.txt
18	Data18.txt
20	Data20.txt
22	Data22.txt
24	Data24.txt
26	Data26.txt

You should assume that that there will be at least 16 cases and at most 26 cases; also, you should assume that the number of cases is even.

### Structure of Data Files

If the player chooses to play a game with 26 cases, the file *Data26.txt* must be opened. The first 26 lines of the file contain the monetary amounts (integer values) that should be assigned to each case. The remaining lines contain the number of rounds for the game and the number of cases to be opened in each round. The contents of *Data16.txt*, *Data18.txt*, *Data20.txt*, *Data22.txt* and *Data24.txt* are organized in the same way.

Contents of Data26.txt	Description
1	 Monetary value corresponding to each of the 26 cases.
5	
10	
15	
25	
50	
75	
100	
200	
300	
400	
500	
750	
1000	
5000	
10000	
25000	
50000	
75000	
100000	
200000	
300000	
400000	
500000	
750000	
1000000	
10	 '10' indicates that there are 10 rounds in the game. 6 cases must be chosen by the player in Round #1, 5 in Round #2, 4 in Round #3, etc.
6	
5	
4	
3	
2	
1	
1	
1	
1	

### Functionality of Program

1. The program should start by requesting the player to choose the number of cases for the game (16, 18, 20, 22, 24, or 26).
2. Based on the number of cases chosen, the appropriate text file is opened and the data is read and stored (preferably in 1-D arrays, one for the cases and one for the rounds). The remainder of this section assumes that the player chooses 26 cases.
3. The player should be requested to choose a case to start the game. This case must be removed from play and set aside, but it must not be opened.
4. In each round of the game, the player is requested to pick a certain number of unopened cases (as specified in the data file), one by one. At the start of the round and before the player is asked to choose a case, the unopened cases must be displayed. For example, if the player chooses case 26 at the beginning, at the start of Round #1, the following cases should be displayed:

We are in Round 1.

You must choose 6 cases in this round.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

As each case is opened, the monetary value is shown and the monetary value for that case is 'removed' (i.e., not shown) from the "money board". For example, suppose that the player picks case 13 as the first case in Round #1. If case 13 has \$15.00, it is 'removed' as follows:

Please choose a case: 13

The value of case 13 is 15. The unopened cases contain the following values

1	1000
5	5000
10	10000
	25000
25	50000
50	75000
75	100000
100	200000
200	300000
300	400000
400	500000
500	750000
750	1000000

When a case is opened, it is also 'removed' from the list of available cases to choose from:

1	2	3	4	5
6	7	8	9	10
11	12		14	15
16	17	18	19	20
21	22	23	24	25

Please choose a case:

5. At the end of each round, **except the last**, the banker makes an offer to the player. The offer is calculated as follows:
  - (a) Find the average of the monetary amount in the unopened cases.
  - (b) Multiply the result in (a) by (number of the current round / total number of rounds), using floating point division.
6. The player can accept the offer and terminate the game ("Deal"). If the player does not accept the offer ("No Deal"), the next round is entered. This continues until there are no more rounds.

### Programming Guidelines

1. You will find it helpful to use three arrays in your program. The first array can be used to store the actual contents of the cases (unknown to the player). The second array can be used to store information on the rounds. The third array can be used to store the cases which will be displayed to the player in order, from 1 to (number of cases selected).
2. The cases which will be displayed to the player are randomly mapped to actual cases using a shuffle algorithm. This algorithm is identical to the one used in Assignment 1 to shuffle responses for multiple-choice questions.
3. The code for playing one round of the game is basically the same as playing any other round. The only difference is the number of cases to be opened by the player in that round.
4. The first case chosen by the player must be treated differently from all the other cases since it is not opened until the end of the game.
5. You must check for invalid cases being chosen by the player (e.g., case numbers outside the possible range, case numbers already opened, or a case number that is the same as the starting case).
6. The file *open* function can use a C-String as an argument. So, you can generate a filename based on the number of cases selected by the player. C-String functions such as *strcpy* and *strcat* should be helpful when creating the filename.

### Implementing “The Twist”

You are free to implement any feature/s for the “twist”. However, feature/s implemented should influence the outcome of the game, i.e., it should not be decorative or simply add sound to the game. It should involve using probabilities and choices that may lead to an alternative outcome. This could happen at any point during the game. It should be interesting and creative. It should utilize any of the concepts covered in the course.

Some examples of possible features are:

- Ask a series of multiple-choice questions depending on the person’s current winnings.
- Ask the player to sort some randomly generated values in order (ascending or descending).
- Ask the player to make a choice from a set of options that have different probabilities of occurring.
- Ask the player to solve a few randomly generated arithmetic problems.

These are just suggestions and you are free to come up with your own ideas.

### Submission Instructions

Please create a zipped file containing the following files as well as **all** the data files:

Assignment4.cpp                      -- Source code for Assignment 4

Assignment4.pdf                      -- PDF document with a short description of “The Twist”

The zipped file should be named [your student ID].zip or [your student ID].rar. These are the **only** file types that will be accepted by the submission system.

The zipped file should not contain .exe or .obj files.