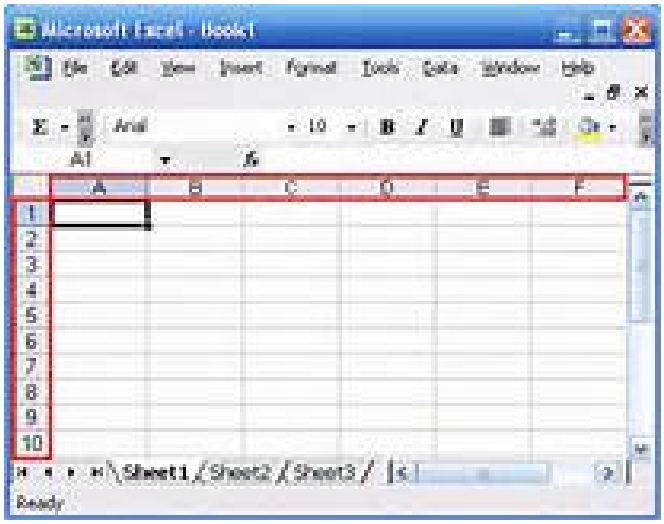
## ITSE 1430- Fall 2016
## Program Set #1

**See C# Grading /Program Guide Sheet for directions and grading/submission information.**

**Note:** Create a console application program for each of the following problems. Do not create a windows application (form).



1. Microsoft Excel is a spreadsheet program. It uses rows and columns to identify cell locations. Rows are numbered with numeric digits while columns are labeled with letters. The first 26 columns are labeled as A through Z. The next 26^2 columns are labeled AA through ZZ and so on. Write a C# program than given an integer column number (`1-1000000`) output its corresponding Excel column name.

**Sample Runs:**

```
Enter a positive column number: 80
The Excel column is: CB

Enter a positive column number: 703
The Excel column is:  AAA
```

Name the program: `ExcelColsXX.cs`, where XX are your initials.

2. In number theory numbers can be classified in many different ways. An evil number is a positive whole number which has an even number of 1's in its binary form. Write a C# program the that determines if a number entered is evil or not. Input a single positive integer (`0-1000`) and outputs the number in binary and determines if the number is "evil" or "not evil". Remember to print the emoticons at the end! The number `0` is considered to be evil. Use at least one user-defined method in your program. Do not list out all the evil numbers- the values must be algorithmically calculated. Output should look similar to below.

**Sample Runs:**

```
Enter a positive number: 15

15 in binary: 0000 1111
15 is an evil number >:)

Enter a positive number: -1
Enter a positive number: 997

997 in binary is 0000 0011 1110 0101
997 is not an evil number :)
```

Name the program: `EvilNosXX.cs`, where XX are your initials.

3. When flying frequently on most airlines, a passenger can join a frequent flyer club. In this problem, benefits are based on the number of segments and miles that a passenger flies. A segment is a one-way trip regardless of the number of stops on the route. A round trip is two segments. Members are classified as Platinum, Gold or Silver Elite Status when they attain a threshold of segments or miles. If these thresholds convey different status, a member is awarded the higher one. The table below shows the threshold number of segments or miles that a member must have to attain the named Elite Status.

| **Elite Status** | **Number of Segments Flown** | **Number of Miles Flown** |
|---|---|---|
| Platinum | 120 or more | 75000 or more |
| Gold | 80-119 | 50000-74999 |
| Silver | 40-79 | 25000-49999 |
| Member | 0-39 | 0-24999 |

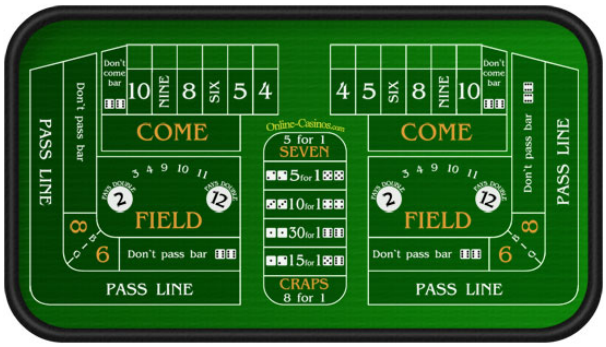
Write a C# program that inputs the number of segments and miles flown. For each segment/mile pair, based on the table above, print only the Elite status of the person with those segments/miles. Check for valid inputs (negative values). Output should look similar to below.

**Sample Runs:**

```
Enter the number of segments flown: 17
Enter the number of miles flown: 24342

Elite status: Member

Enter the number of segments flown: 118
Enter the number of miles flown: 74853

Elite status: Gold

Enter the number of segments flown: 120
Enter the number of miles flown: 53234

Elite status: Platinum
```

Name the program: `AirMilesXX.cs`, where XX are your initials.

4 (**). Write a C# program that simulates the game of craps. Craps is a dice game in which the players make wagers on the outcome of the roll, or a series of rolls, with a pair of dice. The craps table to the right shows the many bets that a player can make.




- **Pass line:** The simplest of these bets is a "pass line" bet. To make a pass line bet, a player places one or more chips on a table in the area marked "pass line". Once all bets are placed, a "shooter" rolls the dice. This is called the "come out roll" roll.
    - If the come out roll shows 7 or 11, the pass line bet wins and the game is over.
    - If the come out roll shows 2, 3, or 12, the pass line bet loses and the game is over.
    - If the come out roll shows 4, 5, 6, 8, 9, or 10, that number is called the "point" and the shooter rolls again and continues to roll until he rolls the point or a 7. If the shooter rolls a 7, the pass line player loses his bet and the game is over. If the shooter rolls the point, the player wins, and the game is over.
- **Don't pass line:** This is the opposite of a pass line bet. That is, the player loses when the pass line bet wins and wins when the pass line bet losses, except if a 12 is rolled on the come out roll. When this occurs, the game is a tie and the player takes back his/her wager.
- **Place bets:** A place bet is made after the point has been established. To make a place bet, place one or more chips on 4, 5, 6, 8, 9, or 10. If your number is rolled before a 7, you win. If a 7 rolled before your number, you lose.
- **Field bets:** A field bet is a one-roll bet. A player bets that the next roll of the dice will be a 2, 3, 4, 9, 10, 11, or 12.

The program should also allow the user to enter an initial amount of money into an account. In addition to placing a bet, the user should be able to specify an amount to go with the bet; this amount should be deducted from the account. Any winnings should be added to the account according to the payout table below; for losses, only deduct (subtract) the bet amount stated above. The updated account value should then be printed out. The user should not be allowed to bet more than the total in the account, and only allowed to bet in 5 dollar increments. Winnings are calculated as follows:

| Bet Type | Payout |
|---|---|
| Pass Line | 1 to 1 |
| Don't Pass Line | 1 to 1 |
| Place-Numbers: 4 or 10 | 9 to 5 |
| Place-Numbers: 5 or 9 | 7 to 5 |
| Place-Numbers: 6 or 8 | 7 to 6 |
| Field- Number: 2 | 2 to 1 |
| Field- Number: 12 | 3 to 1 |
| Field- All other numbers | 1 to 1 |

Next, implement a menu with two options representing different modes for establishing the number. The first option is the test mode- a test mode is required so the program can be graded. In test mode, the user enters both numbers (number and bet). This will allow you to test your logic, results, etc. The second option is game mode. In this mode, the program will randomly generate the number(s) and the user can bet (guess) at the number. Have the user only once enter an initial choice of `G` or `T` mode at start of the program- the mode will remain until the user decides to quit. Output should be user friendly and have error checking.

Name the program: `CrapsGameXX.cs`, where XX are your initials.