Instructions: You can write your project program in any programming language that you'd like. Once finished, turn in **the source code** and **a screenshot of your program running.**

Combinatorics Project

Formulas

Probability in a Bernoulli Trial: Given a simple experiment, called a **Bernoulli trial**, and an event that occurs with a probability p, if the trial is repeated independently n times, then the probability of having exactly k successes is [1]

$$C(n,k) \cdot p^k \cdot (1-p)^{n-k}$$

Expected (Average) Value: For a given probability experiment, let X be a random variable whose possible values come from the set of numbers $x_1, ..., x_n$. Then the **expected value of** X, denoted by $E[X]$, is the sum

$$(x_1) \cdot Prob(X = x_1) + (x_2) \cdot Prob(X = x_2) + ... + (x_n) \cdot Prob(X = x_n)$$

[1]From Discrete Math by Ensley and Crawley, page 460

Program: Calculating the Expected Average

For this program, you will have the user enter information about the trial we're looking at. You will compute the **expected average,** $E[X]$.

You should have a menu in the program like this:

```
1. Enter game info
2. Calculate expected average
3. Quit
```

Keep running the program until the user decides to quit.

Part 1: Getting info about the trials

To help illustrate the data, let's assume the user will be entering data for the 3rd set of example data - tossing 3 coins.

Index i	Value x_i	Events E_i	Event amt $n(E_i)$	Probability $Prob(E_i)$
1	3	{ (H, H, H) }	1	1/8
2	2	{ (H, H, T), (H, T, H), (T, H, H) }	3	3/8
3	1	{ (H, T, T), (T, H, T), (T, T, H) }	3	3/8
4	0	{ (T, T, T) }	1	1/8

You will need to ask the user for $n(S)$ first. In this example, since there are three coins to toss, the total size of the sample set is $2^3 = 8$.

```
What is the size of the sample set, n(S)?
>> 8
```

Next, we need to know how many game outcomes there are. This is different from the sample size. In this case, each Heads gives us +1 point, so there are four outcomes: 3 points, 2 points, 1 point, and 0 points.

```
How many game value outcomes are there?
(How many x[i] items?)
>> 4
```

Then, loop to get all the possible game score outcomes there are. These should be stored in some sort of list structure.

```
x[1] = ?
>> 3
```

We need to store the probability as well, but it would be better to get the event set size $n(E_i)$ and then compute the probability on the program side.

```
How many events lead to this score?
>> 1
```

Probability of getting this value x_1 will be $n(E_1)/n(S)$. Store the probability in a list structure as well.

```
Probability calculated as 0.125%
```

After entering all the game info, it would be useful to display the table of data on the main menu like this:

```
Current game data:

Outcome #      Score value         Probability
0              3                   0.125%
1              2                   0.375%
2              1                   0.375%
3              0                   0.125%
```

Part 2: Computing the expected average For this one, you will need a variable to store the expected value. The expected value is calculated like this:

$$E[X] = \sum_{i=1}^{6} (x_i \cdot Prob(X = x_i))$$

So you will use a loop to get a running sum.

For example...

```
float expectedValue = 0;
for ( int i = 0; i < totalScoreOutcomes; i++ )
{
    expectedValue += values[i] * probability[i];
}
```

Once the loop is completed, you will have $E[X]$. Display it to the screen.

```
Calculated expected value is:  1.5
```

Test out the program with several sets of data. The calculations are given in the following three examples.

Example data

Use these examples to test your program output.

Rolling 1 die:

When rolling one die, the sample size, $n(S)$, is 6. There are 6 total outcomes. In this case, the outcome values match the die roll value, but this isn't always true in all these examples provided.

Value of outcome x_i	Set of events that give the outcome E_i	Amount of events that result in the outcome value $n(E_i)$	Probability of event $Prob(E_i) = \frac{n(E_i)}{n(S)}$
1	{ 1 }	1	1/6
2	{ 2 }	1	1/6
3	{ 3 }	1	1/6
4	{ 4 }	1	1/6
5	{ 5 }	1	1/6
6	{ 6 }	1	1/6

Expected value calculation:

$$E[X] = \sum_{i=1}^{6} (x_i \cdot Prob(X = x_i)$$

$= x_1 \cdot Prob(X = x_1) + x_2 \cdot Prob(X = x_2) + x_3 \cdot Prob(X = x_3) + x_4 \cdot Prob(X = x_4) + x_5 \cdot Prob(X = x_5) + x_6 \cdot Prob(X = x_6)$

$= 1(\frac{1}{6}) + 2(\frac{1}{6}) + 3(\frac{1}{6}) + 4(\frac{1}{6}) + 5(\frac{1}{6}) + 6(\frac{1}{6})$

$= \frac{1+2+3+4+5+6}{6} \qquad = \frac{21}{6} \qquad = 3.5$

The expected value calculated is 3.5.

Rolling 2 dice and summing their values for the outcomes:

We are rolling two dice, so there are $n(S) = 36$ total outcomes that can happen. Since we're summing the values, we will have values of 2 through 12 possible (1+1 up to 6+6).

Index i	Value x_i	Events E_i	Event amt $n(E_i)$	Probability $Prob(E_i)$
1	2	{ (1, 1) }	1	1/36
2	3	{ (1, 2), (2, 1) }	2	2/36
3	4	{ (1, 3), (2, 2), (3, 1) }	3	3/36
4	5	{ (1, 4), (2, 3), (3, 2), (4, 1) }	4	4/36
5	6	{ (1, 5), (2, 4), (3, 3), (4, 2), (5, 1) }	5	5/36
6	7	{ (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1) }	6	6/36
7	8	{ (2, 6), (3, 5), (4, 4), (5, 3), (6, 2) }	5	5/36
8	9	{ (3, 6), (4, 5), (5, 4), (6, 3) }	4	4/36
9	10	{ (4, 6), (5, 5), (6, 4) }	3	3/36
10	11	{ (5, 6), (6, 5) }	2	2/36
11	12	{ (6, 6) }	1	1/36

Expected value calculation:

$$E[X] = \sum_{i=1}^{11} (x_i \cdot Prob(X = x_i)$$

$= 2(\frac{1}{36}) + 3(\frac{2}{36}) + 4(\frac{3}{36}) + 5(\frac{4}{36}) + 6(\frac{5}{36}) + 7(\frac{6}{36}) + 8(\frac{5}{36}) + 9(\frac{4}{36}) + 10(\frac{3}{36}) + 11(\frac{2}{36}) + 12(\frac{1}{36})$

$$= \frac{2 + 6 + 12 + 20 + 30 + 42 + 40 + 36 + 30 + 22 + 12}{36}$$

$$= \frac{252}{36} \qquad = 7$$

The expected value calculated is 7.

Flipping 3 coins, giving a point for heads:

In this game, we flip 3 coins. For each HEADS we get, +1 point is gained. Otherwise, 0 points are gained for tails. For this example, $n(S) = 8$ and there will be 4 total possible values:

Index i	Value x_i	Events E_i	Event amt $n(E_i)$	Probability $Prob(E_i)$
1	3	{ (H, H, H) }	1	1/8
2	2	{ (H, H, T), (H, T, H), (T, H, H) }	3	3/8
3	1	{ (H, T, T), (T, H, T), (T, T, H) }	3	3/8
4	0	{ (T, T, T) }	1	1/8

Expected value calculation:

$$E[X] = \sum_{i=1}^{4}(x_i \cdot Prob(X = x_i)$$

$$= 3(\frac{1}{8}) + 2(\frac{3}{8}) + 1(\frac{3}{8}) + 0(\frac{1}{8})$$

$$= \frac{3+6+3+0}{8} \qquad = \frac{12}{8} \qquad = 1.5$$

The expected value calculated is 1.5.

Example output

Main menu:

```
-----------------------------------------------------
 | Main Menu |
 -------------

Current game data:

Outcome #      Score value         Probability


 1. Enter game info
 2. Calculate expected average
 3. Quit

 >> 1
```

Getting game info:

```
 What is the size of the sample set, n(S)?

 >> 8

 How many game value outcomes are there? (How many x[i]
   items?)

 >> 4

x[1] = 3
How many events lead to this score? 1
Probability calculated as 0.125%.

x[2] = 2
How many events lead to this score? 3
Probability calculated as 0.375%.

x[3] = 1
How many events lead to this score? 3
Probability calculated as 0.375%.
```

```
x[4] = 0
How many events lead to this score? 1
Probability calculated as 0.125%.
```

Main menu after game data entry:

```
-----------------------------------------------------
 | Main Menu |
 -------------

Current game data:

Outcome #      Score value        Probability
0              3                  0.125%
1              2                  0.375%
2              1                  0.375%
3              0                  0.125%


 1. Enter game info
 2. Calculate expected average
 3. Quit
```

Computing expected average:

```
E[X] = 3 * 0.125 + 2 * 0.375 + 1 * 0.375 + 0 * 0.125

Calculated expected value is: 1.5
```

Turn in

Turn in your source file(s) and a screenshot of your program running.