

Introduction to Java

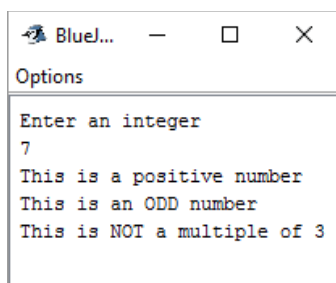
Unit 3. Programs

Start a new project for this unit.

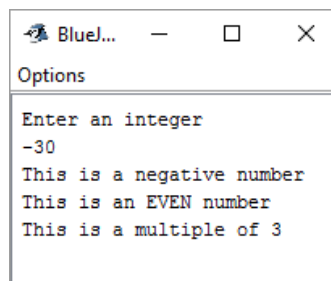
Programs 1 to 4 require you to use the mod, %, operator.

1. Let the user enter an integer. Your program should then say whether the number is:
positive, negative, or zero
even or odd
a multiple of 3 or not

Below are two examples of how the program should run.



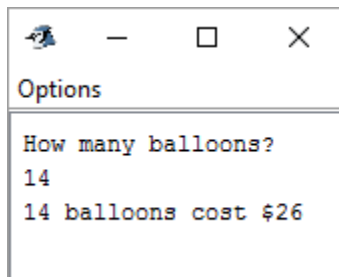
```
Options
Enter an integer
7
This is a positive number
This is an ODD number
This is NOT a multiple of 3
```



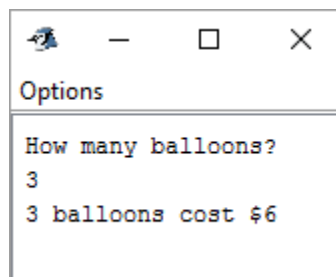
```
Options
Enter an integer
-30
This is a negative number
This is an EVEN number
This is a multiple of 3
```

2. You sell balloons for \$2 but \$9 for a pack of five. Write a program where the user enters the number of balloons and the program prints the total cost. Solve this without using an *if* statement.

Below are two examples of how the program should run.



```
Options
How many balloons?
14
14 balloons cost $26
```



```
Options
How many balloons?
3
3 balloons cost $6
```

3. Write a program that generates a random integer between 10 (inclusive) and 1,000 (inclusive). This represents a random number of seconds. Then convert the seconds into minutes and seconds. Here are some sample outputs:

271 seconds = 4 minutes and 31 seconds

46 seconds = 0 minutes and 46 seconds

780 seconds = 13 minutes and 0 seconds

Solve this without using an *if* statement.

4. Write a program where the user enters a particular year and the program either says “This is a leap year” or it says “This is not a leap year.”

Here are the rules for determining if a year is a leap year or not:

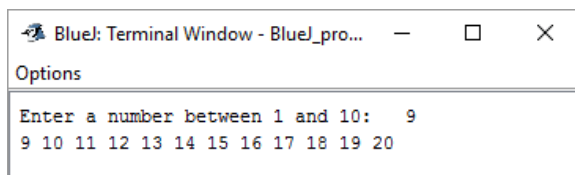
- Any year divisible by 400 is always a leap year.
- Any year divisible by 100 is not a leap year (unless it is also divisible by 400).
- Any year divisible by 4 is a leap year (unless it is also divisible by 100 and not 400)
- All other years are not leap years.

For example: 1900 and 1901 were not leap years; 2000 and 2004 were leap years.

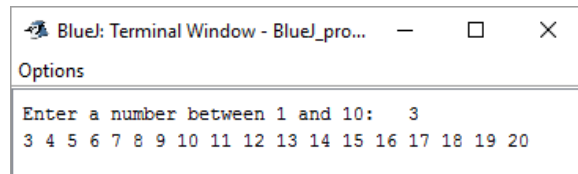
Programs 5 to 9 involve the use of while loops.

5. Write a program where the user enters a number between 1 and 10 and the program displays all the integers from that number up to and including 20.

Below are two examples of how the program should run.



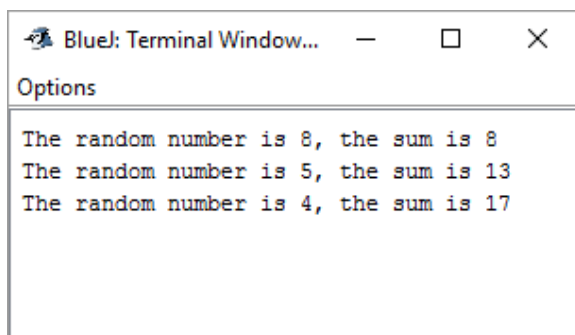
```
Blue: Terminal Window - Blue_pro...  
Options  
Enter a number between 1 and 10: 9  
9 10 11 12 13 14 15 16 17 18 19 20
```



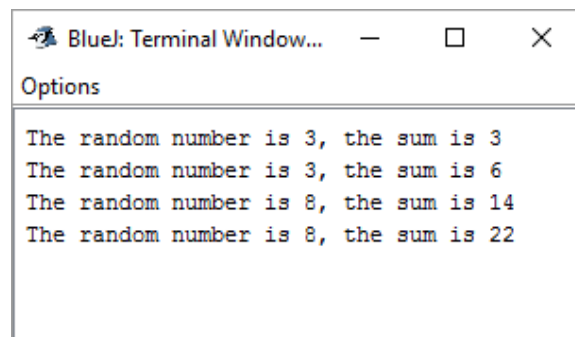
```
Blue: Terminal Window - Blue_pro...  
Options  
Enter a number between 1 and 10: 3  
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

6. Write a program that displays random integers between 3 (inclusive) and 8 (inclusive). The program should stop after the sum of the random numbers is 17 or greater. Create one variable for the random number and a second variable to hold the sum. After generating the random number, add it to the sum.

Below are two examples of how the program should run.



```
Blue: Terminal Window...  
Options  
The random number is 8, the sum is 8  
The random number is 5, the sum is 13  
The random number is 4, the sum is 17
```



```
Blue: Terminal Window...  
Options  
The random number is 3, the sum is 3  
The random number is 3, the sum is 6  
The random number is 8, the sum is 14  
The random number is 8, the sum is 22
```

7. In this program we will build a better version of 21 than we did in Unit 2. The player can take as many cards as they want. The dealer will keep taking cards until their total is 17 or greater. To the right are some sample runs.

As a reminder, the rules of the game are:

The player and dealer are both trying to get as close to 21 as possible without going over.

The player goes first and keeps taking cards until they decide to stop. In our initial version of the game the player can take as many cards as they want. You should fix this so that they cannot take anymore cards once they get to 21 or more.

The dealer keeps taking cards until they have 17 or more. You should fix this so that the dealer does not take any cards if the player has gone over 21 ("busted").

The winner is decided as follows:

- If the player goes over 21, they lose
- If the player has 21 or less and the dealer has more than 21, player wins
- If the player and dealer both have 21 or less, the one with the higher total wins
- It is a tie if both players have 21 or less and have the same totals.

Here is the beginning of the program where the player gets their cards.

```
Scanner input = new Scanner( System.in );
int card1 = (int)(10*Math.random() ) + 1;
int card2 = (int)(10*Math.random() ) + 1;
int player = card1 + card2;
System.out.println( "You have " + card1 + " and " + card2 + " for " + player );
System.out.print( "Enter 1 for another card, any other number to stay. " );
int response = input.nextInt();
while ( response == 1 ){
    int another_card = (int)(10*Math.random() ) + 1;
    player += another_card;
    System.out.println( "You got a " + another_card + " for a total of " + player );
    System.out.print( "Enter 1 for another card, any other number to stay. " );
    response = input.nextInt();
}
```

You write the code for the dealer and to figure out who won or lost. You should also fix the above code so that the player cannot keep taking cards forever.

```
BlueJ: Terminal Window - BlueJ_programs
Options
You have 3 and 10 for 13
Enter 1 for another card, any other number to stay. 1
You got a 2 for a total of 15
Enter 1 for another card, any other number to stay. 1
You got a 2 for a total of 17
Enter 1 for another card, any other number to stay. 4

Dealer got a 7 for a total of 7
Dealer got a 5 for a total of 12
Dealer got a 8 for a total of 20
Player loses
```

```
BlueJ: Terminal Window - BlueJ p SMART Ink
Options
You have 9 and 3 for 12
Enter 1 for another card, any other number to stay. 1
You got a 1 for a total of 13
Enter 1 for another card, any other number to stay. 1
You got a 7 for a total of 20
Enter 1 for another card, any other number to stay. 5

Dealer got a 2 for a total of 2
Dealer got a 4 for a total of 6
Dealer got a 8 for a total of 14
Dealer got a 8 for a total of 22
Dealer busted and lost
```

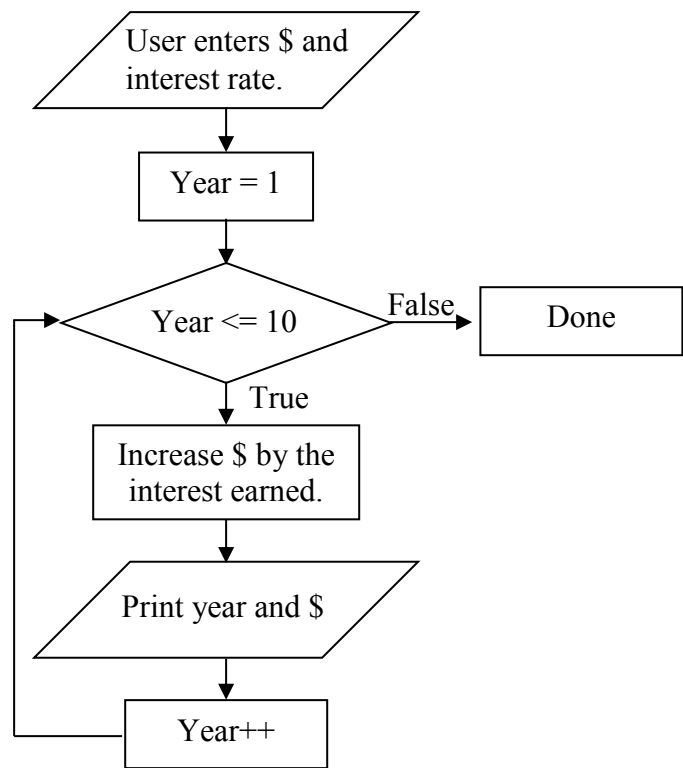
8. Write a program that projects how a person's money will grow over 10 years. Use the flowchart to the right. Below is an examples of how the program should run if the user enters 1000 and 0.1 (10%).

```

Blue: Termina...
Options
How much money are you investing?
1000
What is the interest rate?
0.1

Year 1: $1100.0
Year 2: $1210.0
Year 3: $1331.0
Year 4: $1464.1000000000001
Year 5: $1610.5100000000002
Year 6: $1771.5610000000004
Year 7: $1948.7171000000005
Year 8: $2143.5888100000006
Year 9: $2357.9476910000008
Year 10: $2593.742460100001

```



9. Write a program where the user has to guess a secret number (between 1 and 100). If the guess is wrong, the program says if the guess was too high or too low. After correctly guessing the number, the program says how many guesses it took.

While writing the code you may want to print out the secret random number so that you can verify that program is working correctly. Then at the end you remove that print statement.

To the right is an example of how it should run.

```

Blue: Terminal Wind...
Options
Guess a number
60
Too low
Guess a number
82
Too low
Guess a number
95
Too high
Guess a number
91
Too high
Guess a number
88
Too low
Guess a number
89
Congratulations. It took you 6 tries

```

Problems 10 to 14 should use for loops.

10. Write a program where the user enters two integers, n_1 and n_2 , and the program prints out all the numbers from n_1 to n_2 (inclusive). Assume the user enters a smaller number followed by a larger number. Use a for-loop

Here are some sample runs.

```
Blue: Termin... - □ ×
Options
Enter two integers
-2 9
-2 -1 0 1 2 3 4 5 6 7 8 9
```

```
Blue: Termin... - □ ×
Options
Enter two integers
3 8
3 4 5 6 7 8
```

11. The user enters a number between 1 and 11 and the program prints the first twelve multiples of the number. You must use a for loop.

Here are two examples.

```
Blue: Terminal ... - □ ×
Options
Enter a positive integer
3
3 6 9 12 15 18 21 24 27 30 33 36
```

```
Blue: Terminal ... - □ ×
Options
Enter a positive integer
8
8 16 24 32 40 48 56 64 72 80 88 96
```

12. Write a program that displays the results shown to the right.

Your program should:

- Use a for loop
- Use `Math.pow`
- Use `(int)` to cast the value returned by `Math.pow` to an `int`.

Your code inside the main method should be about four lines long.

```
Blue: Terminal ... - □ ×
Options
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024
2^11 = 2048
2^12 = 4096
2^13 = 8192
2^14 = 16384
2^15 = 32768
```

13. Write a program where the user enters a number and the program displays all the factors of that number. You must use a for-loop.

Below are two sample outputs.

```

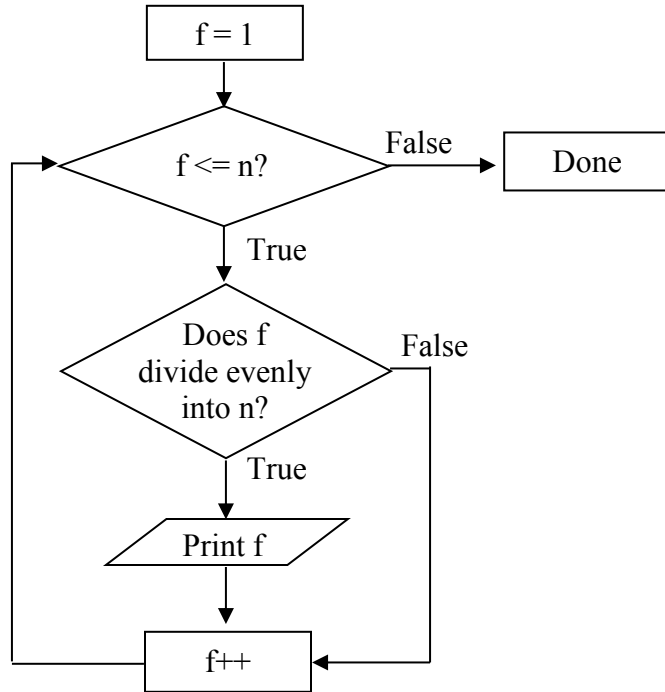
Blue: T... - □ ×
Options
Enter a positive integer
72

The factors of 72 are:
1 2 3 4 6 8 9 12 18 24 36 72
    
```

```

Blue: T... - □ ×
Options
Enter a positive integer
31

The factors of 31 are:
1 31
    
```



14. The user enters a distance (in miles) and the program displays how long it will take to travel there if the driver maintains an average speed of 25, 30, 35, .. 70 miles per hour.

See the figure to the right for a sample output.

This problem tends to be harder than it looks.

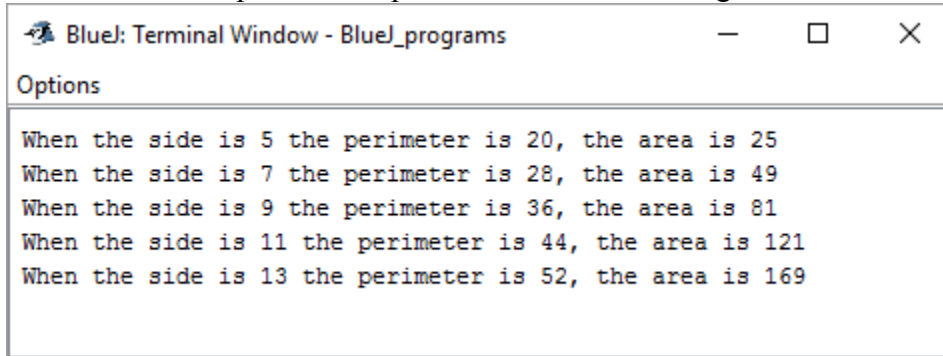
Suggestion: Calculate the time in minutes and cast the result to an integer. Then convert the minutes into hours and minutes.

```

Blue: Terminal Window - Blu... - □ ×
Options
How many miles?
200
To travel 200 miles
at a speed of 25 mph takes 8 hr and 0 min.
at a speed of 30 mph takes 6 hr and 40 min.
at a speed of 35 mph takes 5 hr and 42 min.
at a speed of 40 mph takes 5 hr and 0 min.
at a speed of 45 mph takes 4 hr and 26 min.
at a speed of 50 mph takes 4 hr and 0 min.
at a speed of 55 mph takes 3 hr and 38 min.
at a speed of 60 mph takes 3 hr and 20 min.
at a speed of 65 mph takes 3 hr and 4 min.
at a speed of 70 mph takes 2 hr and 51 min.
    
```

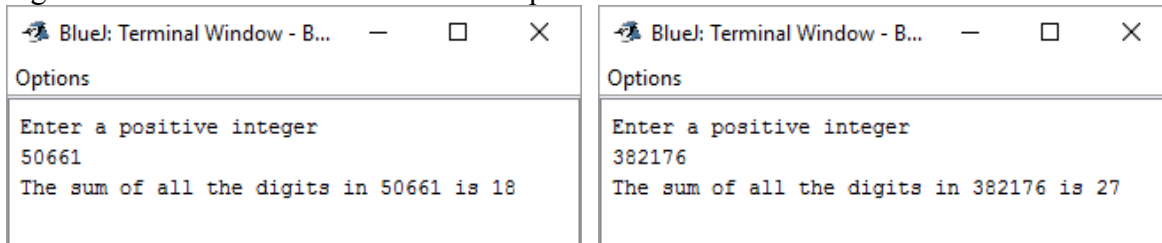
For the remaining problems, you decide what kind of loop to use.

15. Write a program that prints out the perimeter and area of a square for sides of 5, 7, 9, 11, and 13. Use a loop. Your output should look something like this.



```
BlueJ: Terminal Window - BlueJ_programs
Options
When the side is 5 the perimeter is 20, the area is 25
When the side is 7 the perimeter is 28, the area is 49
When the side is 9 the perimeter is 36, the area is 81
When the side is 11 the perimeter is 44, the area is 121
When the side is 13 the perimeter is 52, the area is 169
```

16. Write a program where the user enters a positive integer and the program adds up the digits in the number. Here are two examples.



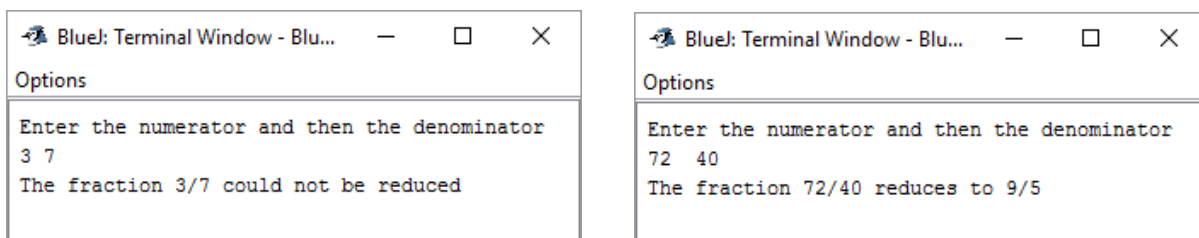
```
BlueJ: Terminal Window - B...
Options
Enter a positive integer
50661
The sum of all the digits in 50661 is 18

BlueJ: Terminal Window - B...
Options
Enter a positive integer
382176
The sum of all the digits in 382176 is 27
```

Hint. Suppose the user enters 835. Notice that
 $835 \% 10$ equals 5 and $835 / 10$ equals 83
 $83 \% 10$ equals 3 and $83 / 10$ equals 8
 $8 \% 10$ equals 8 and $8 / 10$ equals 0

Yes, you need a number. And remember when testing, don't put in a number greater than 2,147,483,647 or your program goes Boom (metaphorically speaking).

17. Write a program where the user enters a numerator and a denominator and the program reduces the fraction (if possible). Here are some sample runs.



```
BlueJ: Terminal Window - Blu...
Options
Enter the numerator and then the denominator
3 7
The fraction 3/7 could not be reduced

BlueJ: Terminal Window - Blu...
Options
Enter the numerator and then the denominator
72 40
The fraction 72/40 reduces to 9/5
```

Hint: Write a loop that looks for the largest number that divides evenly into both the numerator and the denominator. Afterward the loop if the largest common factor is one, then you know the fraction could not be reduced. Otherwise you found the greatest common factor and you can reduce the fraction.