



The art of Vincent Sheridan  
<https://vincentsheridan.com>

Starling Flight 1

# Starlings With a Sense

In this quest, you get to play with branching statements. You will create 6 functions, each of which does something that requires making decisions.

Each of these functions is worth varying numbers of points. It's up to you to find out how many points you can score by making the most progress you can.

As usual, I will give you template code you can copy and flesh out.

## Your first miniquest - Mean of three

To get past this checkpoint you must correctly implement the following function in the template code:

```
double mean_of_3(int n1, int n2, int n3);
```

When I invoke this method, I will supply it three integer parameters. It must return their average value to me as a double.

## Your second miniquest - Max of five

You must implement:

```
int max_of_5(int n1, int n2, int n3, int n4, int n5);
```

When I invoke this method, I will supply it five integer params. It must return the greatest of them to me.

## Your third miniquest - Min of five

You must implement:

```
int min_of_5(int n1, int n2, int n3, int n4, int n5);
```

When I invoke this method, I will supply it five integer params. It must return the least of them to me.

## Your fourth miniquest - Triangle from sides

You must implement:

```
bool sides_make_triangle(int a, int b, int c);
```

When I invoke this method, I will supply it three integer params. Return whether it is possible to form a triangle using the given numbers as side lengths. True means yes.

## Your fifth miniquest - Triangle from angles

You must implement:

```
bool angles_make_triangle(int A, int B, int C);
```

When I invoke this method, I will supply it three integer params. Return whether it is possible to form a triangle using the given numbers as angles in degrees. True means yes.

## Your sixth miniquest - Leap Test

You must implement:

```
bool is_a_leap_year(int year);
```

When I invoke this method, I will supply it an integer parameter. Return whether the given number is a leap year or not (according to the definition you can find in Wikipedia). True means yes.

## Starter code

```
// Student ID: 12345678
// Replace the number above with your Student ID
//
// Branching-Functions.cpp
// 2a-Lab-03
//
#include <iostream>

// This function returns the mean the three numbers passed
// in as parameters. Note that the mean may not be a round
// number. So you must use the double datatype for it.
double mean_of_3(int n1, int n2, int n3) {
    // TODO - YOUR CODE HERE
}

// This function returns the maximum of the 5 given numbers
int max_of_5(int n1, int n2, int n3, int n4, int n5) {
    // TODO - YOUR CODE HERE
}

// This function returns the minimum of the 5 given numbers
int min_of_5(int n1, int n2, int n3, int n4, int n5) {
    // TODO - YOUR CODE HERE
}

// Given three lengths, this function should return whether they can be the
// sides of some triangle. The heuristic you code should check if the
// sum of the two smallest sides is greater than or equal to the third side.
// Treat extreme cases as valid triangles. E.g. a+b == c means valid triangle.
// The challenge is to do it without using arrays
bool sides_make_triangle(int a, int b, int c) {
    // TODO - YOUR CODE HERE
}

// Given three angles as integer degrees, this function should return whether
// they can be internal angles of some triangle. Treat extreme cases as
// valid triangles. E.g. (0, 0, 180) is a valid triangle
bool angles_make_triangle(int A, int B, int C) {
    // TODO - YOUR CODE HERE
}

// Return true if the year yyyy is a leap year and false if not.
bool is_a_leap_year(int yyyy) {
    // TODO - YOUR CODE HERE
}
```

## Testing your own code

You should test your functions using your own `main()` function in which you try and call your functions in many different ways and cross-check their return value against your hand-computed results. But when you submit you must NOT submit your `main`. I will use my own and invoke your functions in many creative ways. Hopefully you've thought of all of them.

### Note

When you define your functions in one file and invoke them from another, the compiler has no way of telling if the number and type of parameters with which you're invoking them are correct. It can't *have your back*.

To help the compiler help you (yes, those who help others help them usually end up getting the most help), you can tell it in advance what the signatures of these functions you will use from elsewhere are. These function signatures are usually either:

- put at the top of the `cpp` file in which you plan to invoke them or
- collectively put within a header file that is then `#included` in the `cpp` source file

In this quest, simply put the below declarations at the top of your `main.cpp` file (or anywhere else you intend to call your functions from). Starting with the next quest, you will put them in a separate header file (and thus need to upload it together with your `cpp` file at the testing site).

```
// Forward declarations of functions that will be used in this file
// before their definitions are encountered by the compiler

double mean_of_3(int n1, int n2, int n3);
int max_of_5(int n1, int n2, int n3, int n4, int n5);
int min_of_5(int n1, int n2, int n3, int n4, int n5);
bool sides_make_triangle(int a, int b, int c);
bool angles_make_triangle(int A, int B, int C);
bool is_a_leap_year(int year);
```

## Submission

When you think you're happy with your code and it passes all your own tests, it is time to see if it will also pass mine.

1. Head over to <https://quests.nonlinearmedia.org>
1. Enter the secret password for this quest in the box
2. Drag and drop your `Branching_Functions.cpp` file into the button and press it.
3. Do not submit your source file that contains your `main()` function. If any of your files contains a `main()`, it must be commented out.
4. Wait for me to complete my tests and report back (usually a minute or less).



### Points and Extra Credit Opportunities

I monitor the discussion forums closely and award extra credit points for well-thought out and helpful discussions.

May the best coders win. That may just be all of you.

Happy Hacking,

&