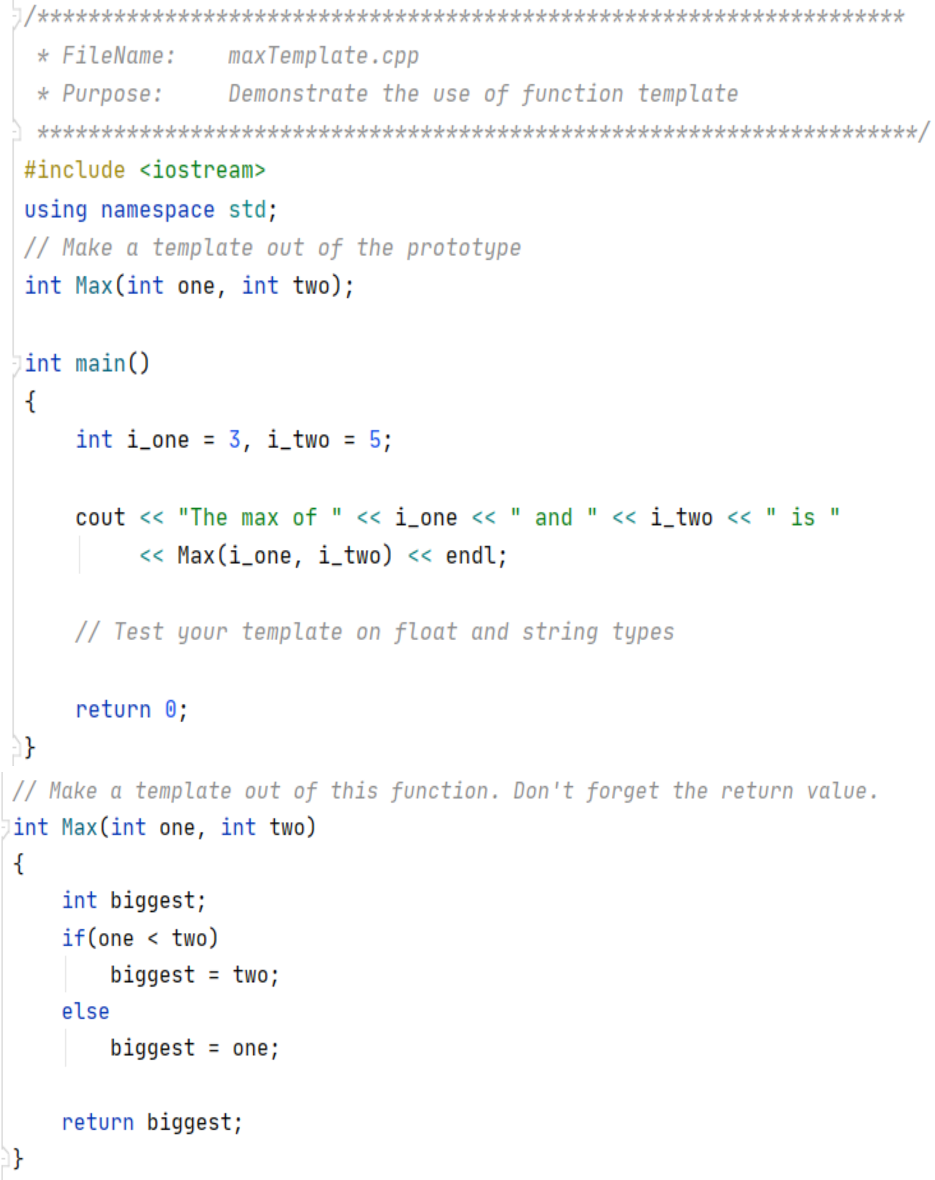
3 Exercises

1. Use the **Max** function to create a template and test it on different data types.

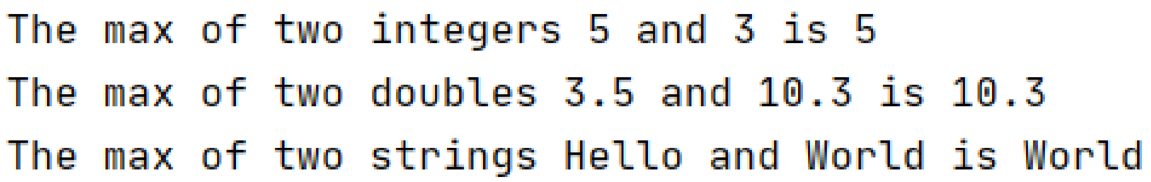
- Copy **maxTemplate.cpp** to your program



```cpp
/*************************************************************************
 * FileName:    maxTemplate.cpp
 * Purpose:     Demonstrate the use of function template
 *************************************************************************/
#include <iostream>
using namespace std;
// Make a template out of the prototype
int Max(int one, int two);

int main()
{
    int i_one = 3, i_two = 5;

    cout << "The max of " << i_one << " and " << i_two << " is "
         << Max(i_one, i_two) << endl;

    // Test your template on float and string types

    return 0;
}
// Make a template out of this function. Don't forget the return value.
int Max(int one, int two)
{
    int biggest;
    if(one < two)
        biggest = two;
    else
        biggest = one;

    return biggest;
}
```

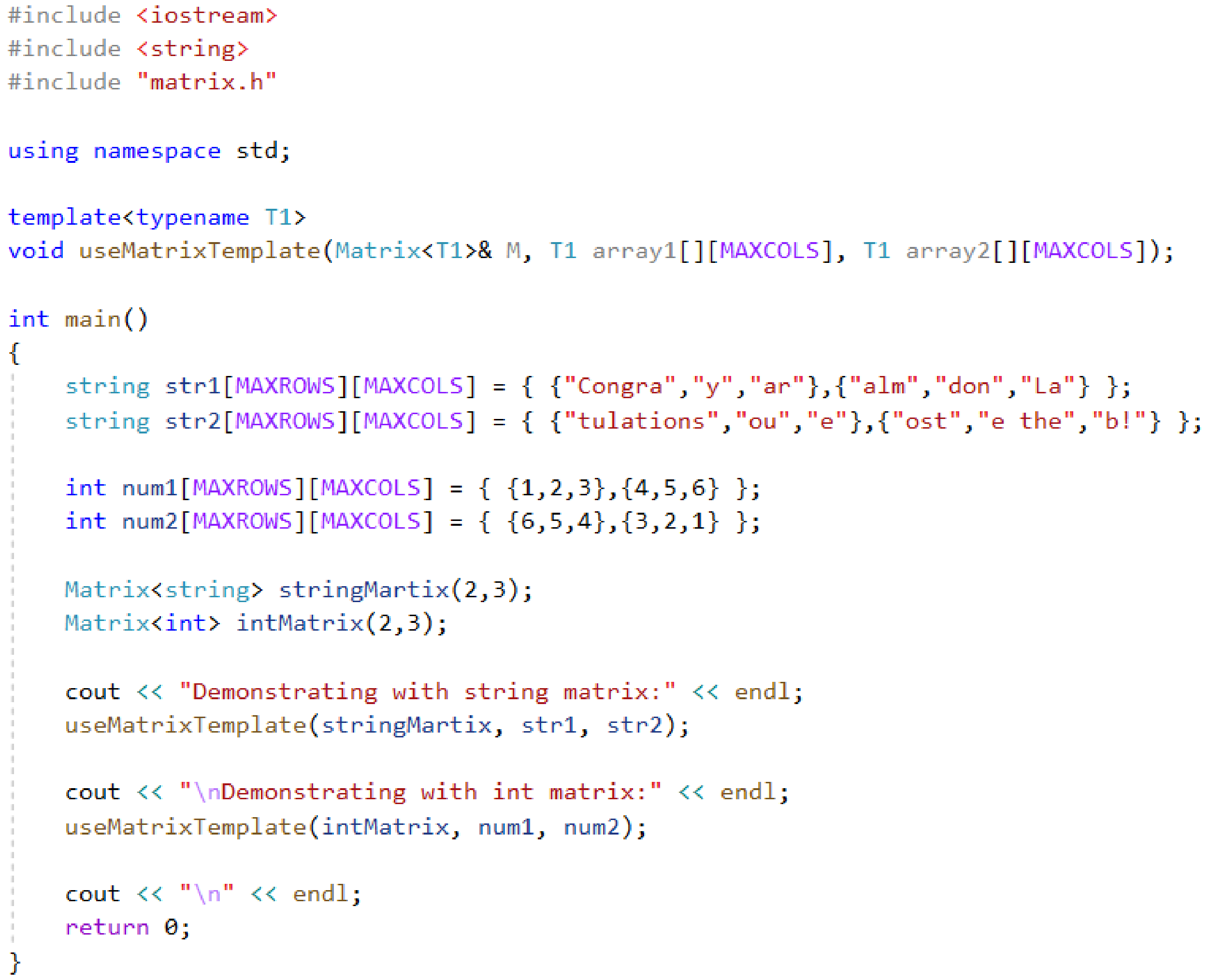
- Compile and run the program to see how it works
- Make a template out of Max. Don’t forget the return type
- Modify the prototype appropriately
- Test your Max template on **int**, **double**, and **string** types.

When you are done, your outputs should resemble this:

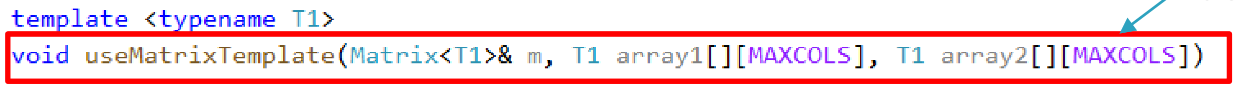


```
The max of two integers 5 and 3 is 5
The max of two doubles 3.5 and 10.3 is 10.3
The max of two strings Hello and World is World
```

2. Complete the Matrix template types and the function **useMatrixTemplate**, make the program run as the sample.



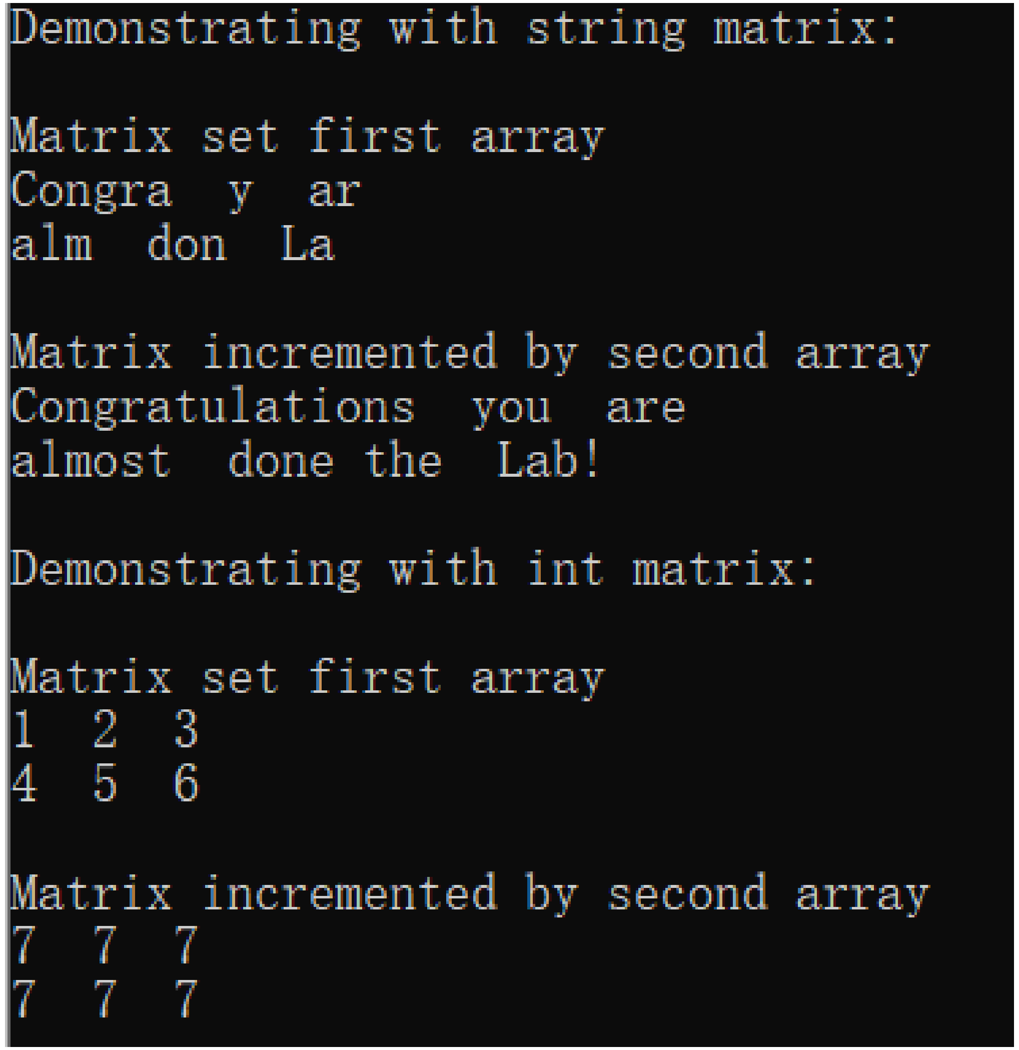
```
#include <iostream>
#include <string>
#include "matrix.h"

using namespace std;

template<typename T1>
void useMatrixTemplate(Matrix<T1>& M, T1 array1[][MAXCOLS], T1 array2[][MAXCOLS]);

int main()
{
    string str1[MAXROWS][MAXCOLS] = { {"Congra","y","ar"},{"alm","don","La"} };
    string str2[MAXROWS][MAXCOLS] = { {"tulations","ou","e"},{"ost","e the","b!"} };

    int num1[MAXROWS][MAXCOLS] = { {1,2,3},{4,5,6} };
    int num2[MAXROWS][MAXCOLS] = { {6,5,4},{3,2,1} };

    Matrix<string> stringMartix(2,3);
    Matrix<int> intMatrix(2,3);

    cout << "Demonstrating with string matrix:" << endl;
    useMatrixTemplate(stringMartix, str1, str2);

    cout << "\nDemonstrating with int matrix:" << endl;
    useMatrixTemplate(intMatrix, num1, num2);

    cout << "\n" << endl;
    return 0;
}
```



complement the function definition

```
template <typename T1>
void useMatrixTemplate(Matrix<T1>& m, T1 array1[][MAXCOLS], T1 array2[][MAXCOLS])
```

Here is a sample of running the program:



```
Demonstrating with string matrix:

Matrix set first array
Congra  y  ar
alm  don  La

Matrix incremented by second array
Congratulations  you  are
almost  done the  Lab!

Demonstrating with int matrix:

Matrix set first array
1  2  3
4  5  6

Matrix incremented by second array
7  7  7
7  7  7
```