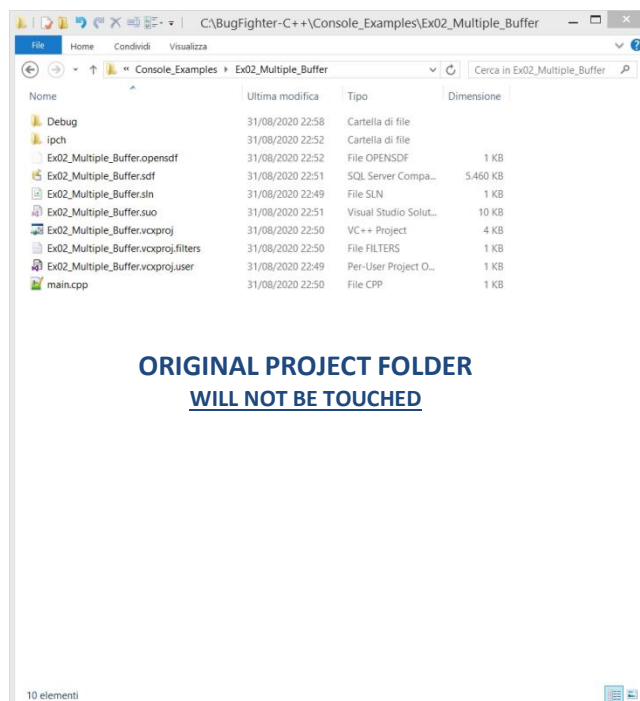
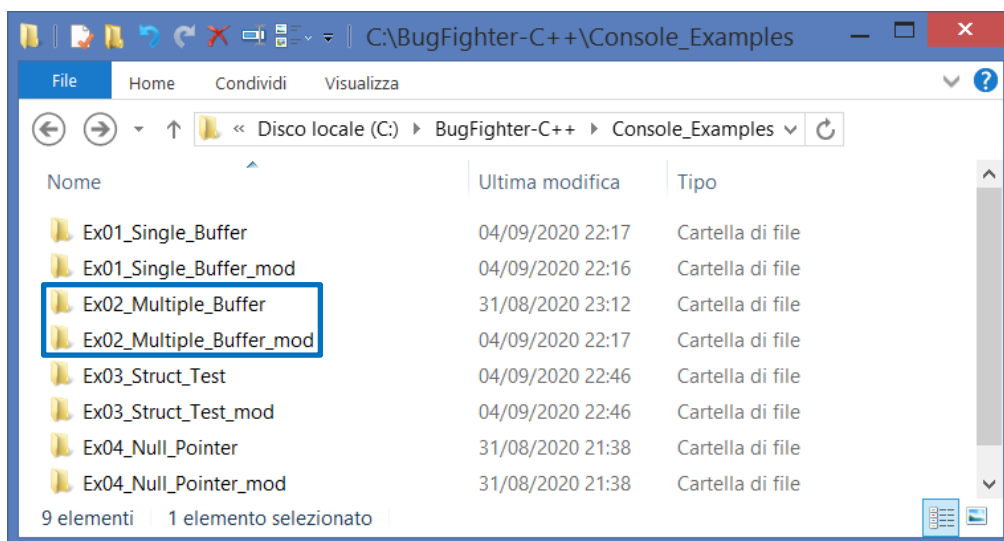


Bugfighter C/C++

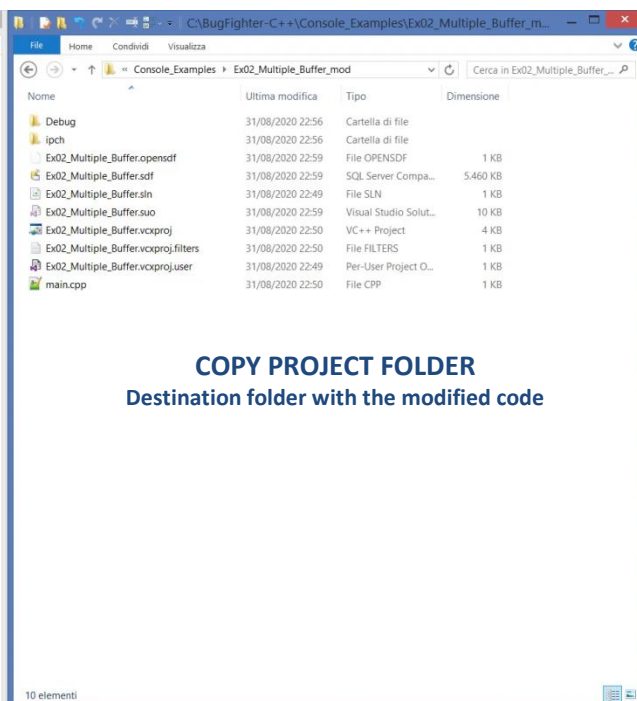
How to detect a buffer overflow in an array

1. Make a copy of the project directory:

Ex: `Ex02_Multiple_Buffer` → `Ex02_Multiple_Buffer_mod`

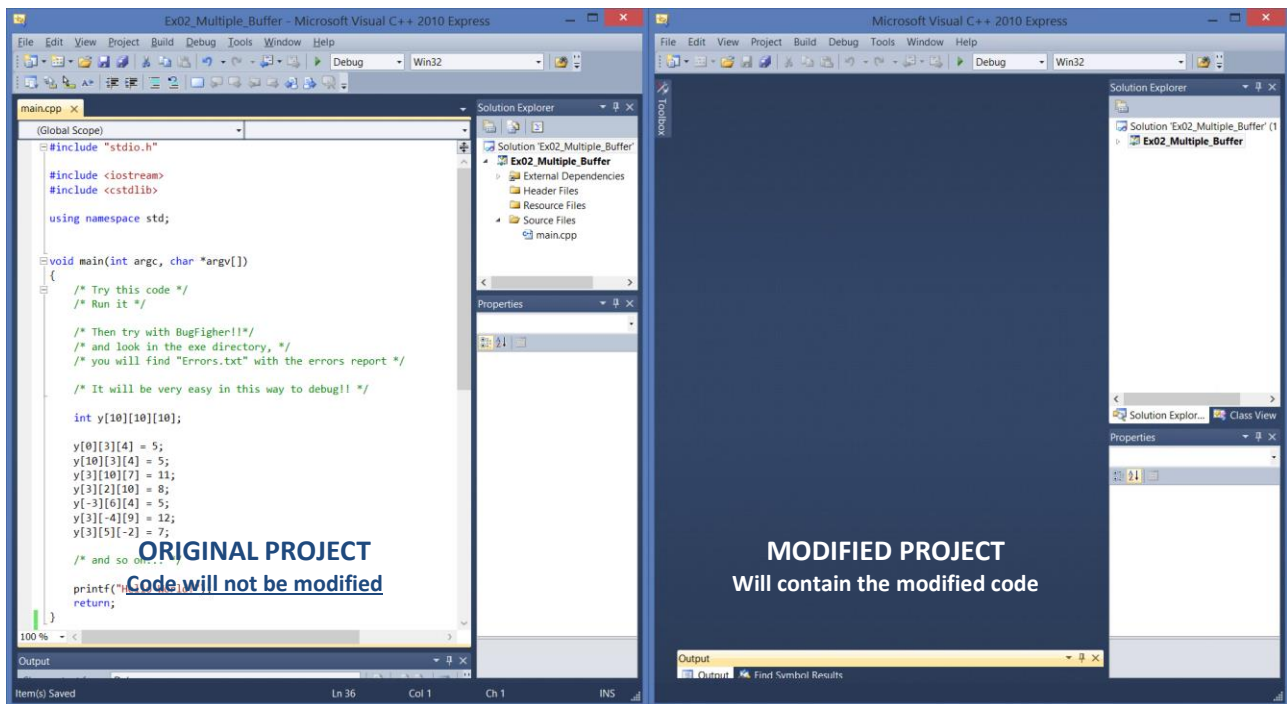


ORIGINAL PROJECT FOLDER
WILL NOT BE TOUCHED



COPY PROJECT FOLDER
Destination folder with the modified code

2. Open both projects with your compiler



```
#include "stdio.h"
```

```
#include <iostream>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
void main(int argc, char *argv[])  
{
```

```
    /* Try this code */  
    /* Run it */
```

```
    /* Then try with BugFigher!!*/  
    /* and look in the exe directory, */  
    /* you will find "Errors.txt" with the errors report */
```

```
    /* It will be very easy in this way to debug!! */
```

```
    int y[10][10][10];
```

```
    y[0][3][4] = 5;  
    y[10][3][4] = 5;  
    y[3][10][7] = 11;  
    y[3][2][10] = 8;  
    y[-3][6][4] = 5;  
    y[3][-4][9] = 12;  
    y[3][5][-2] = 7;
```

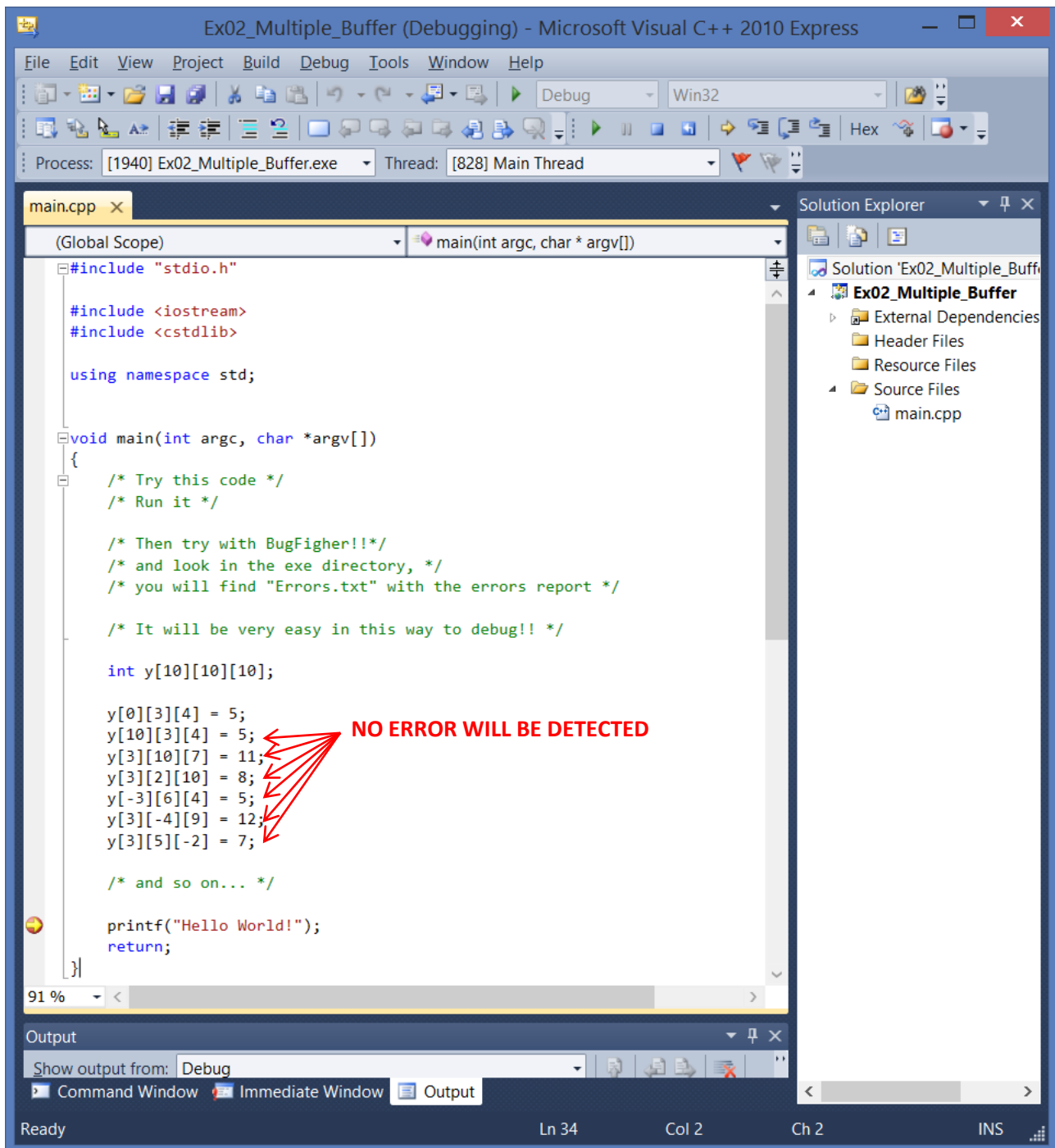
```
    /* and so on... */
```

```
    printf("Hello World!");  
    return;
```

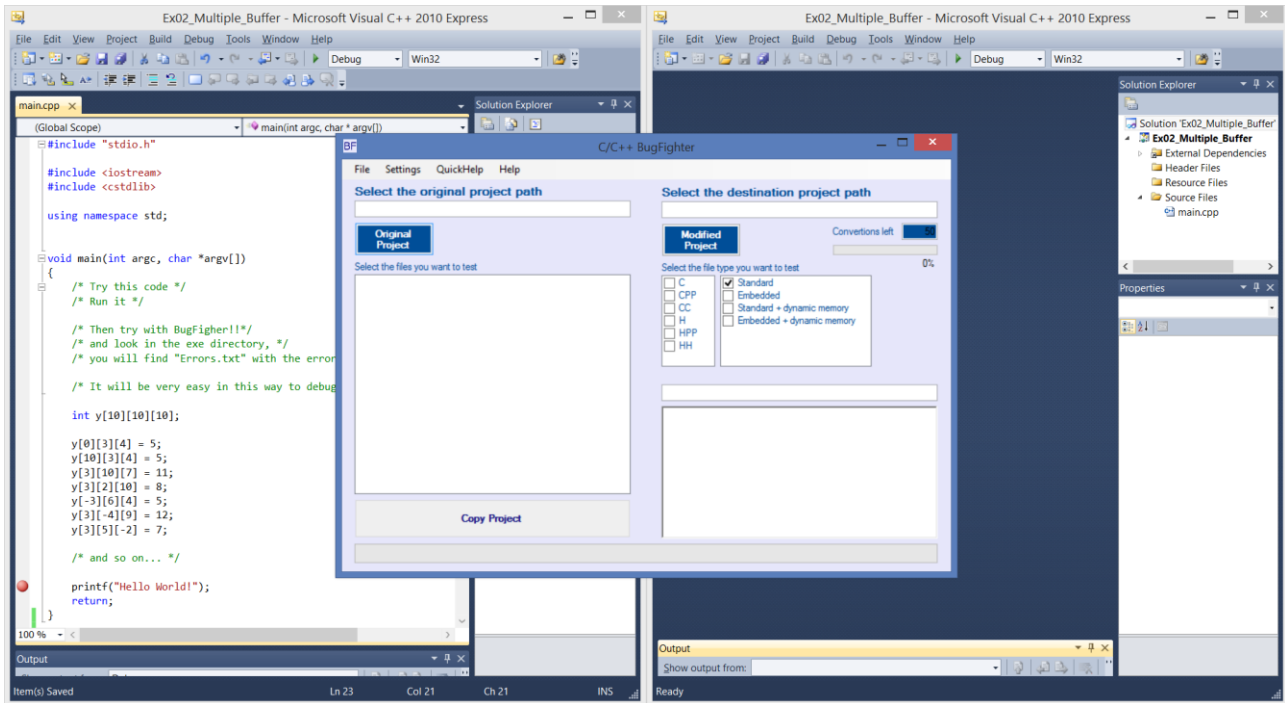
```
}
```

3. Run the original code:

No exception will be detected

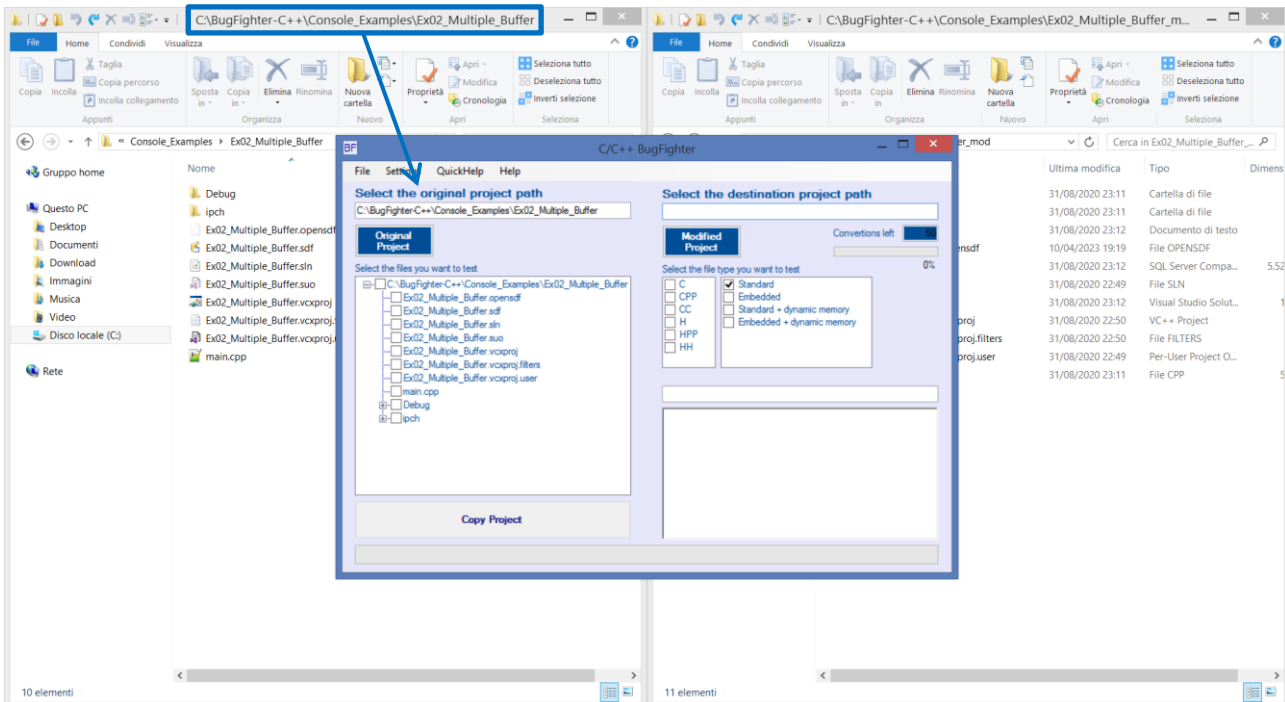


4. Open Bugfighter C/C++



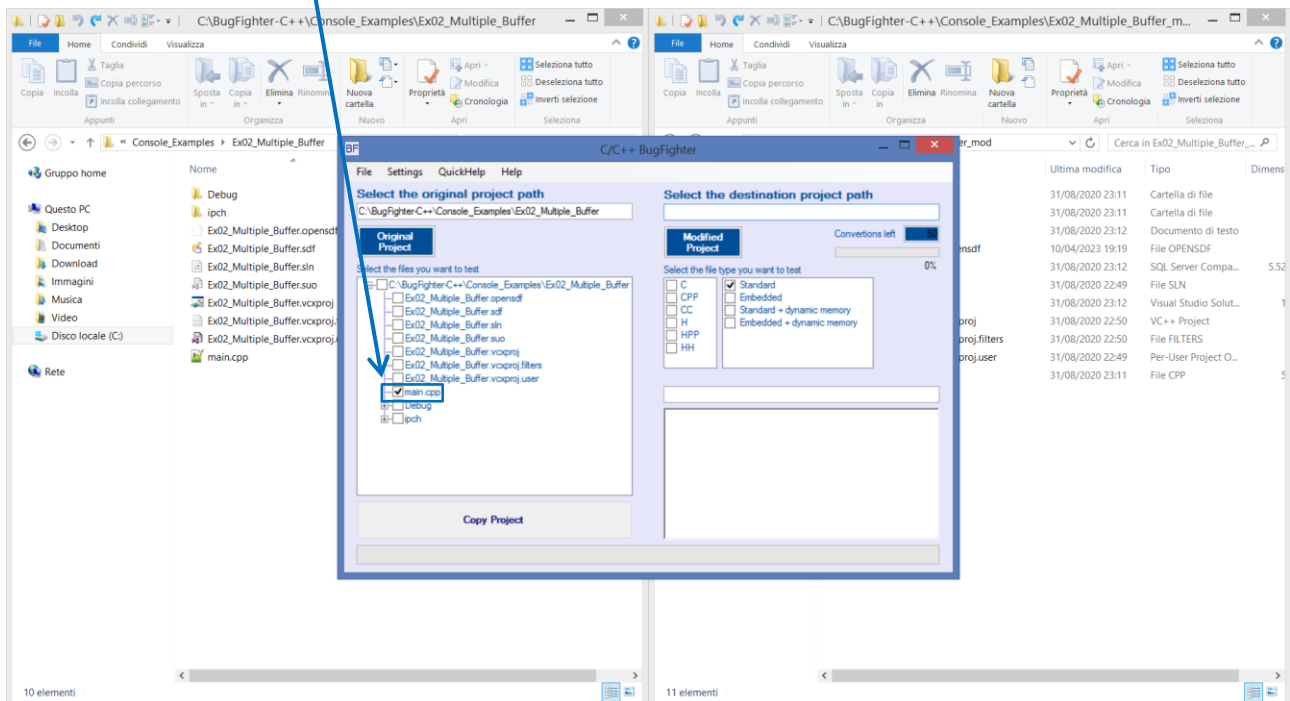
5. Select the directory with the original code:

Ex: "C:\BugFighter-C++\Console_Examples\Ex02_Multiple_Buffer"



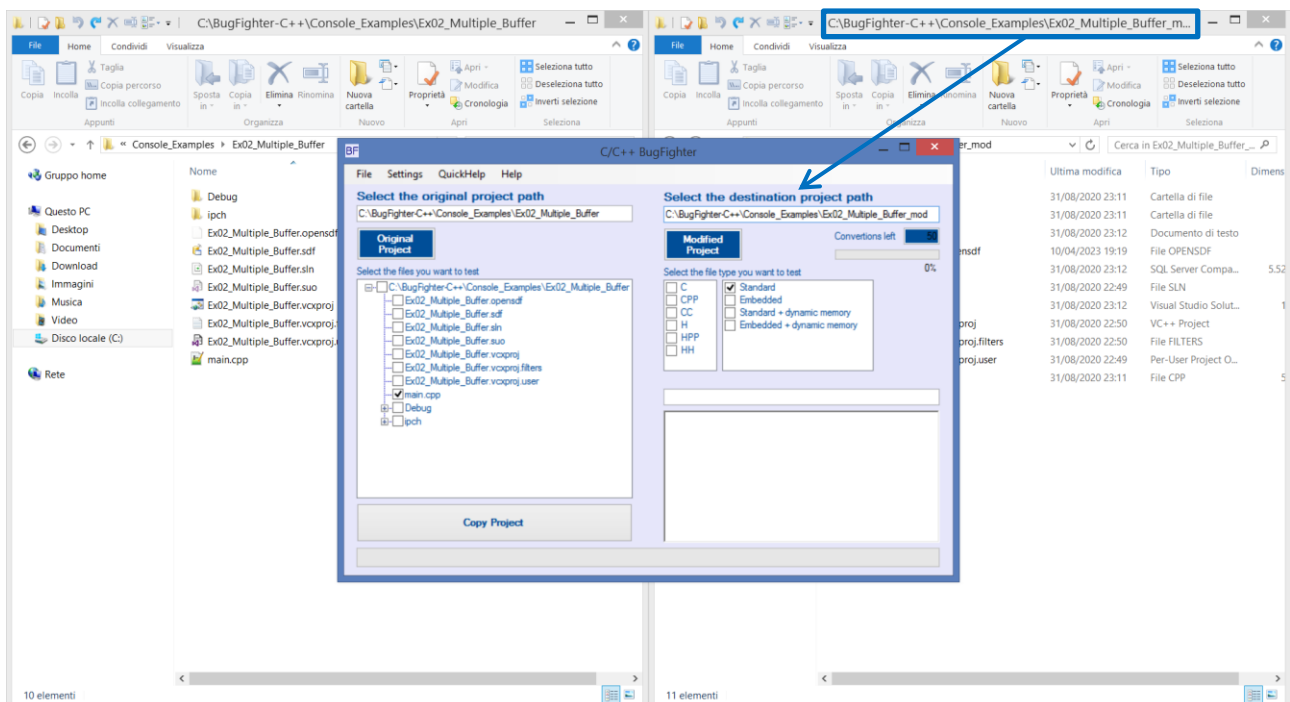
6. Select the C/C++ code file that must be tested:

Ex: "main.cpp"



7. Select the destination directory with the modified code:

Ex: "C:\BugFighter-C++\Console_Examples\Ex02_Multiple_Buffer_mod"

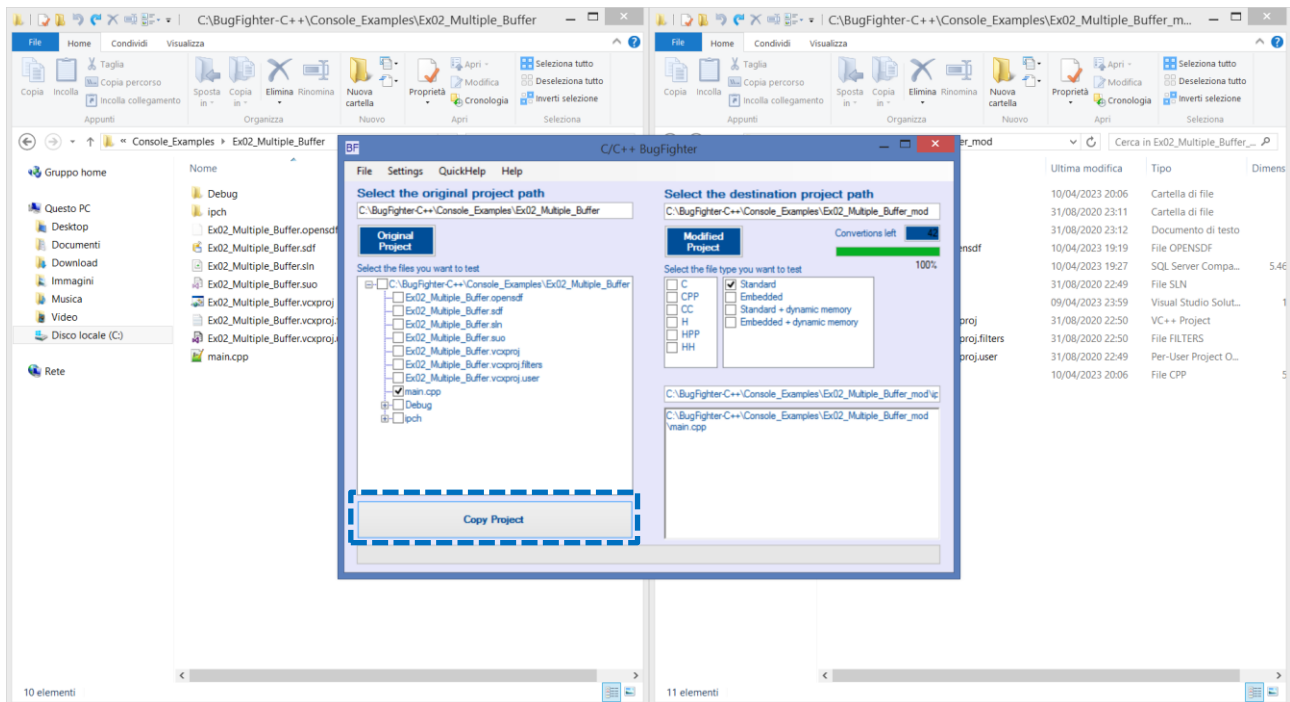


8. Select “Copy Project”:

All files will be copied from original directory to the destination directory:

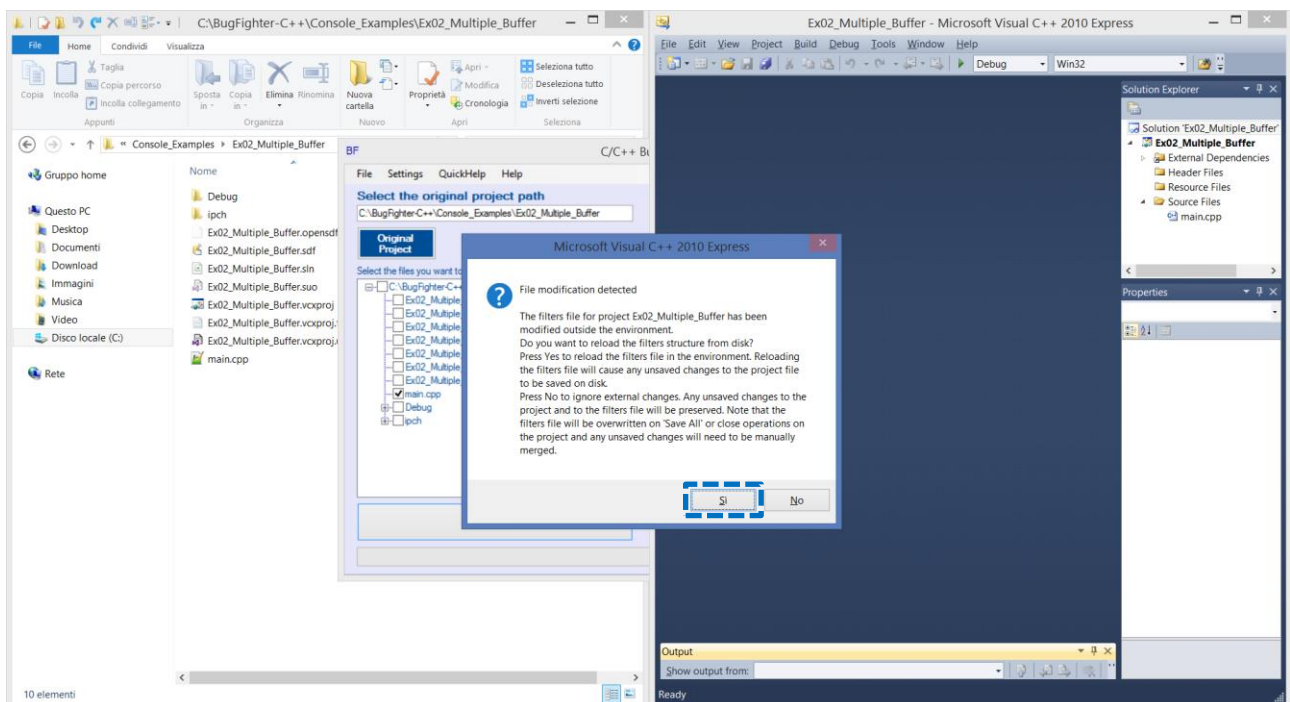
Ex02_Multiple_Buffer → Ex02_Multiple_Buffer_mod

The file “main.cpp” will contain the test code.

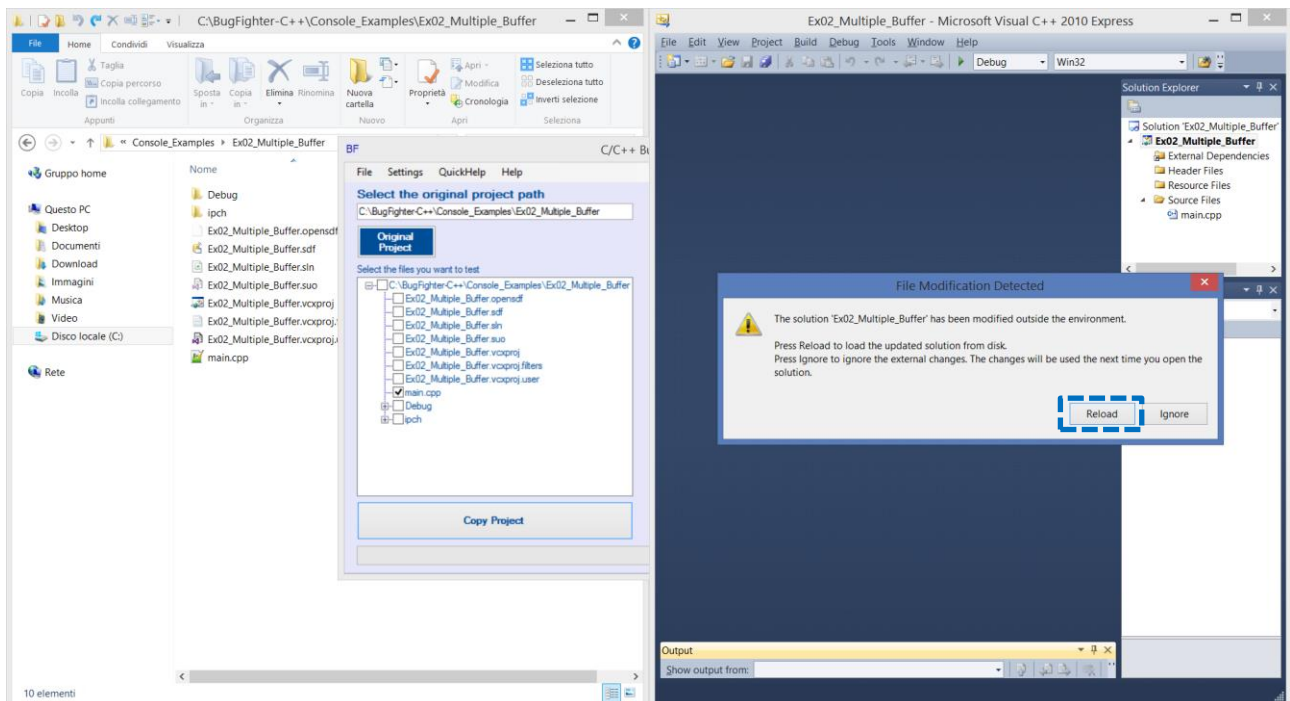


9. The destination project has been updated:

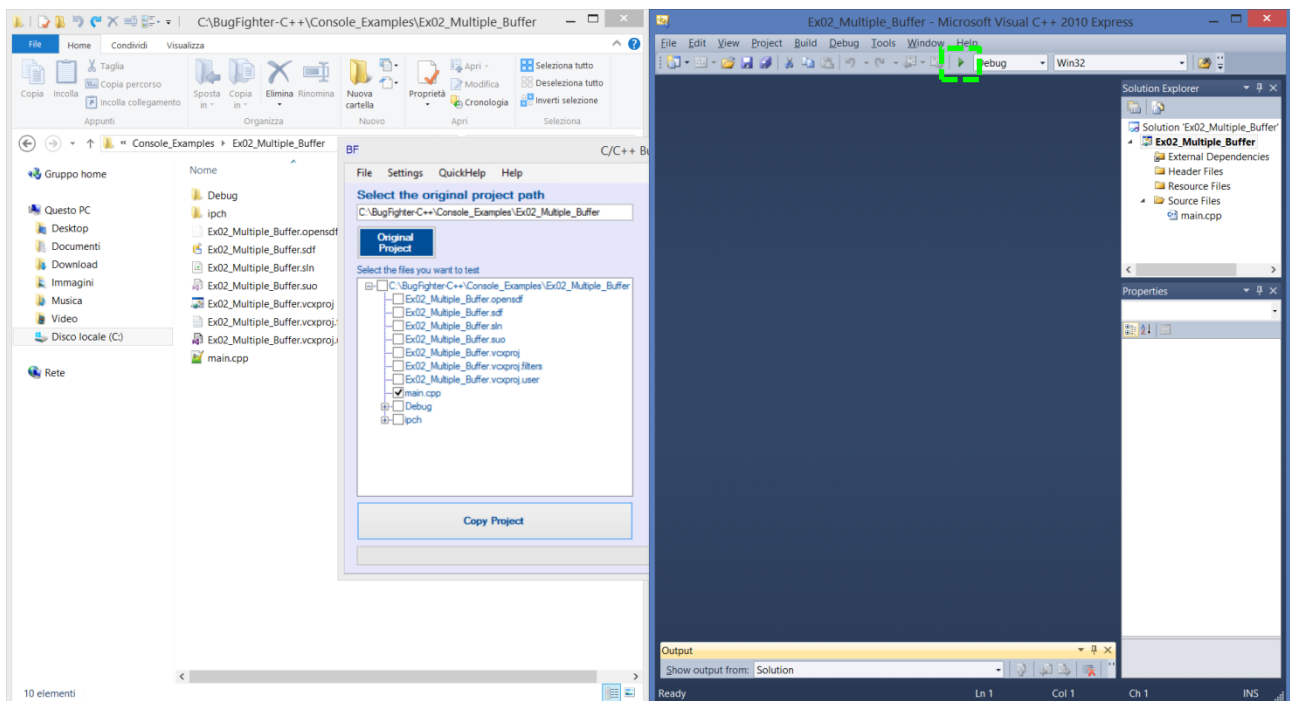
Select “Yes” to reload the project



10. Select “Reload”



11. Select “Run”



12. During run-time the test code will detect the error and write the message on the console:

Error n. 1

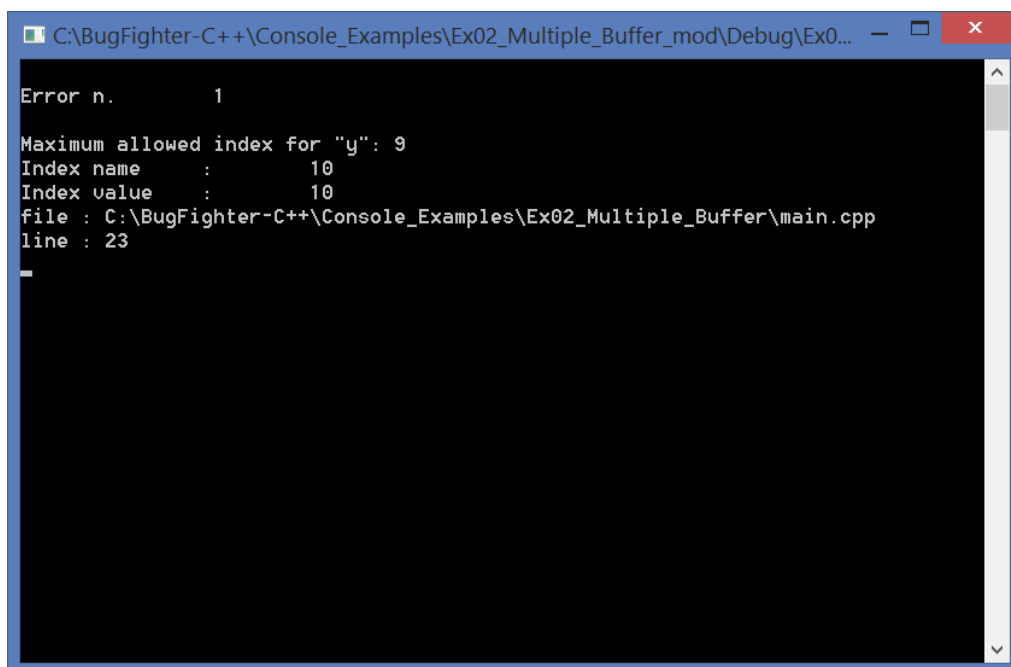
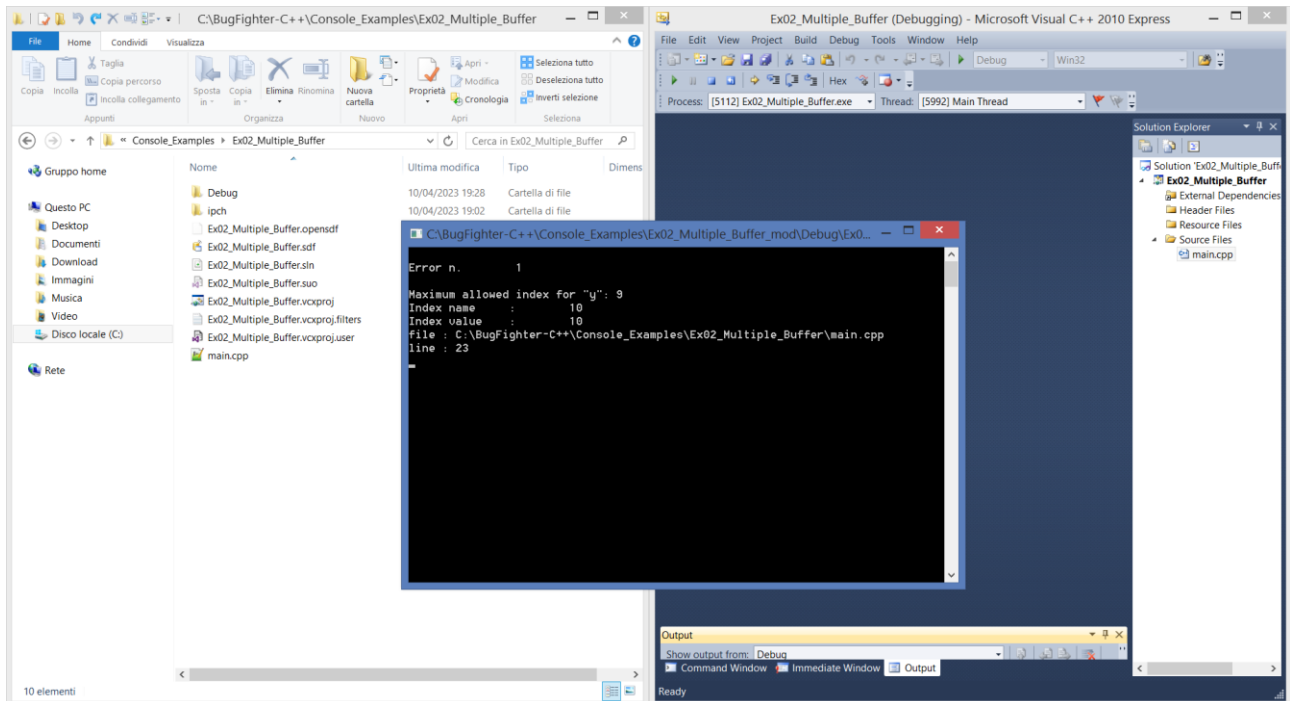
Maximum allowed index for "y": 9

Index name : 10

Index value : 10

file : C:\BugFighter-C++\Console_Examples\Ex02_Multiple_Buffer\main.cpp

line : 23



13. During run-time the test code will detect the error and write the message in a text file "Errors.txt":

Error n. 1

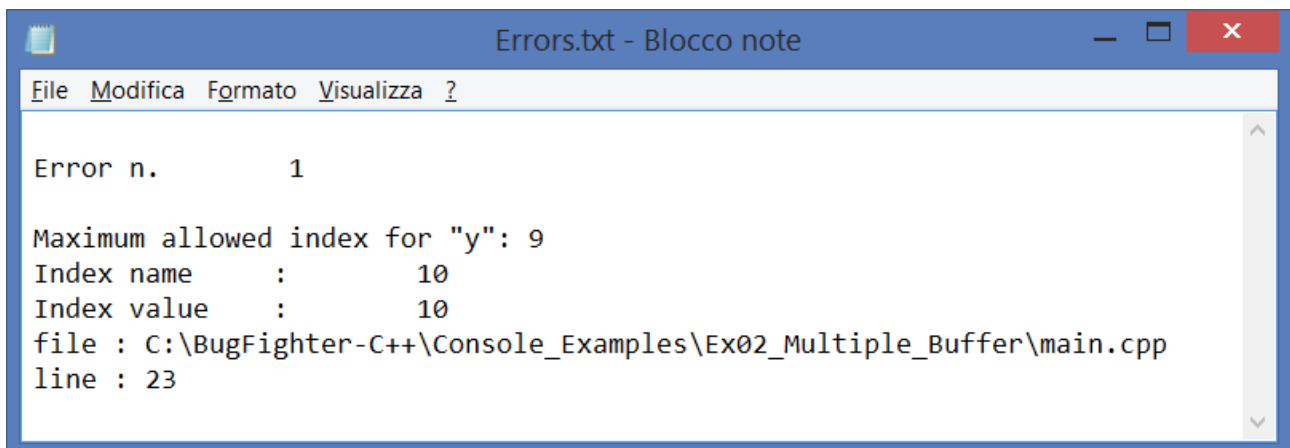
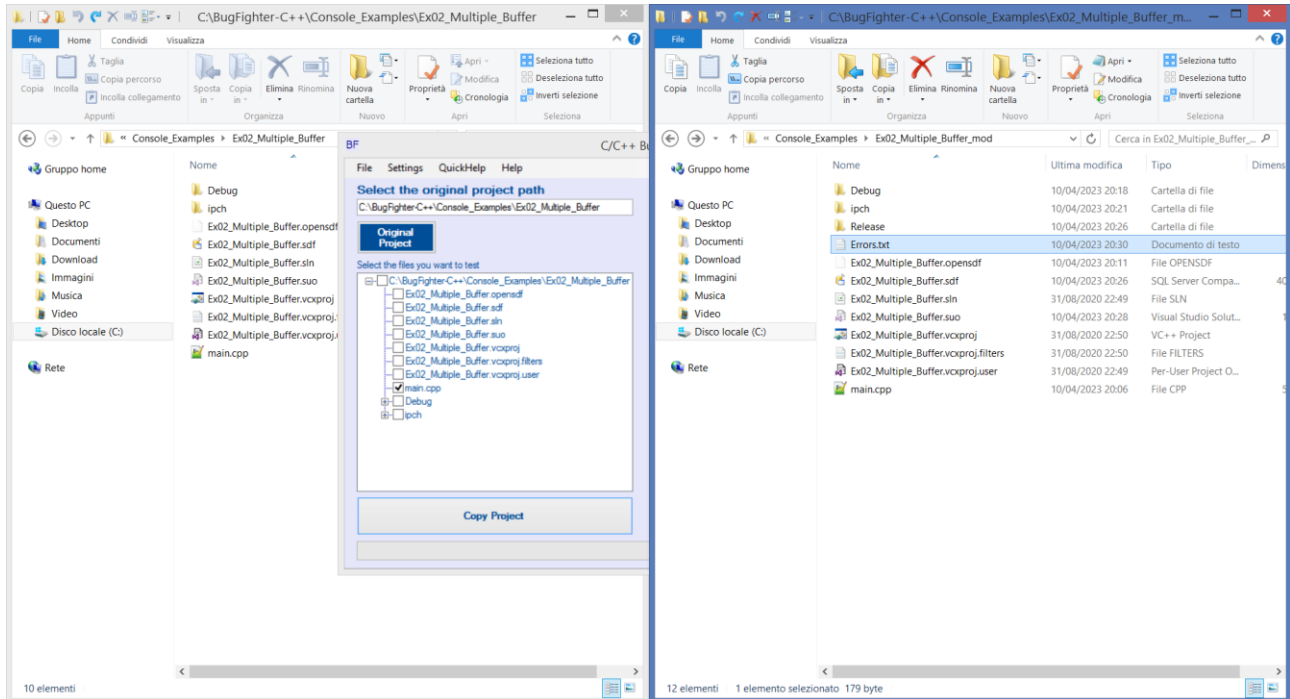
Maximum allowed index for "y": 9

Index name : 10

Index value : 10

file : C:\BugFighter-C++\Console_Examples\Ex02_Multiple_Buffer\main.cpp

line : 23



14. Check "Errors.txt"

The image shows two windows from Microsoft Visual C++ 2010 Express. The top window, titled "Ex02_Multiple_Buffer", displays the source code of `main.cpp`. The code defines a 3D array `y` of size `10x10x10` and performs several assignments. A green arrow points from the `#include <stdio.h>` line to the "Output" window at the bottom. A red arrow points from the `y[10][3][4] = 5;` line to the "Errors.txt" window below. A blue arrow points from the `y[10][3][4] = 5;` line to the "Ln 23" label in the status bar.

```
#include "stdio.h"
#include <iostream>
#include <stdlib>

using namespace std;

void main(int argc, char *argv[])
{
    /* Try this code */
    /* Run it */

    /* Then try with Bugfighter!! */
    /* and look in the exe directory, */
    /* you will find "Errors.txt" with the errors report */

    /* It will be very easy in this way to debug!! */

    int y[10][10][10];

    y[0][3][4] = 5;
    y[10][3][4] = 5;
    y[5][10][7] = 11;
    y[2][2][10] = 8;
    y[-3][6][4] = 5;
    y[3][-4][9] = 12;
    y[3][5][-2] = 7;

    /* and so on... */

    printf("Hallo World!");
    return;
}
```

The bottom window, titled "Errors.txt - Blocco note", displays the error message:

```
Error n.      1

Maximum allowed index for "y": 9
Index name    : 10
Index value   : 10
file : C:\BugFighter-C++\Console_Examples\Ex02_Multiple_Buffer\main.cpp
line : 23
```

15. Correct the index

The screenshot shows the Microsoft Visual C++ 2010 Express IDE. The main window displays the file `main.cpp` with the following code:

```
#include "stdio.h"

#include <iostream>
#include <cstdlib>

using namespace std;

void main(int argc, char *argv[])
{
    /* Try this code */
    /* Run it */

    /* Then try with BugFigher!! */
    /* and look in the exe directory, */
    /* you will find "Errors.txt" with the errors report */

    /* It will be very easy in this way to debug!! */

    int y[10][10][10];

    y[0][3][4] = 5;
    y[9][4][4] = 5;
    y[3][10][7] = 11;
    y[3][2][10] = 8;
    y[-3][6][4] = 5;
    y[3][-4][9] = 12;
    y[3][5][-2] = 7;

    /* and so on... */

    printf("Hello World!");
    return;
}
```

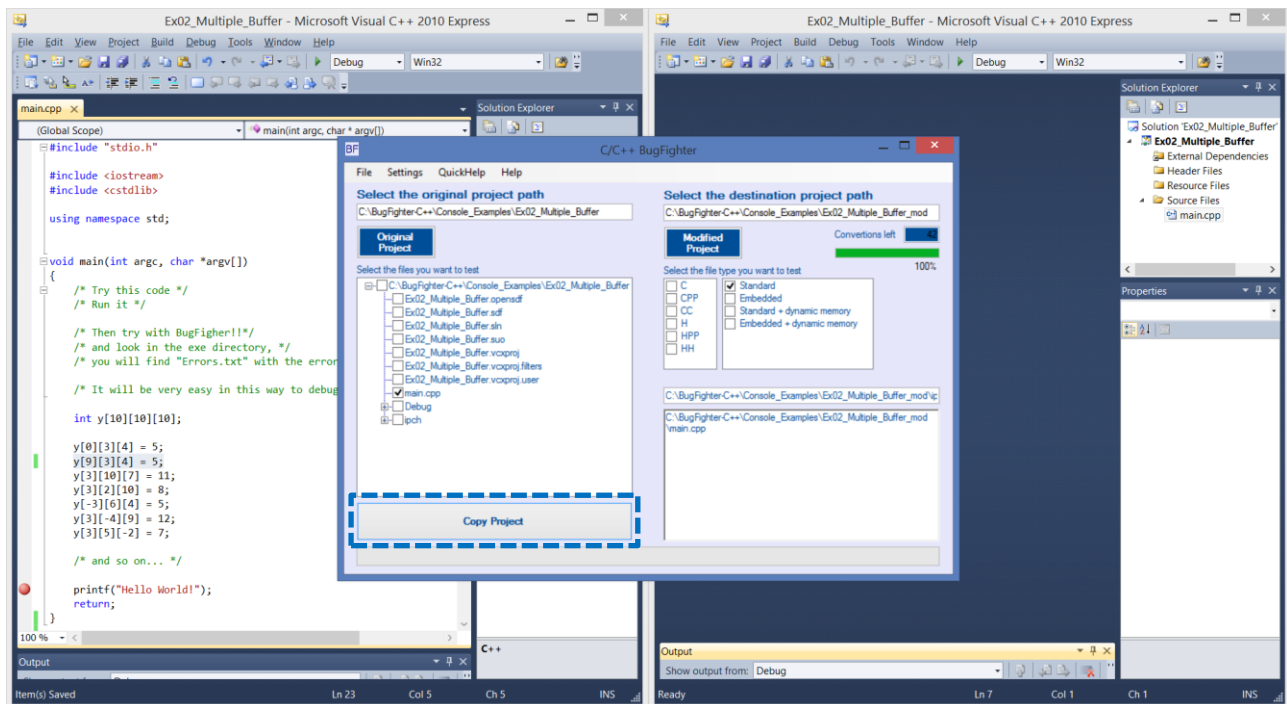
A red arrow points to the line `y[9][4][4] = 5;` with the text **CORRECT THE INDEX**. Below the arrow, a text box shows the correction: `y[10][3][4] → y[9][3][4]`. The `y[10][3][4]` part is highlighted in red in the original image.

The Solution Explorer on the right shows the project structure for 'Ex02_Multiple_Buffer', including 'External Dependencies', 'Header Files', 'Resource Files', and 'Source Files' (containing `main.cpp`).

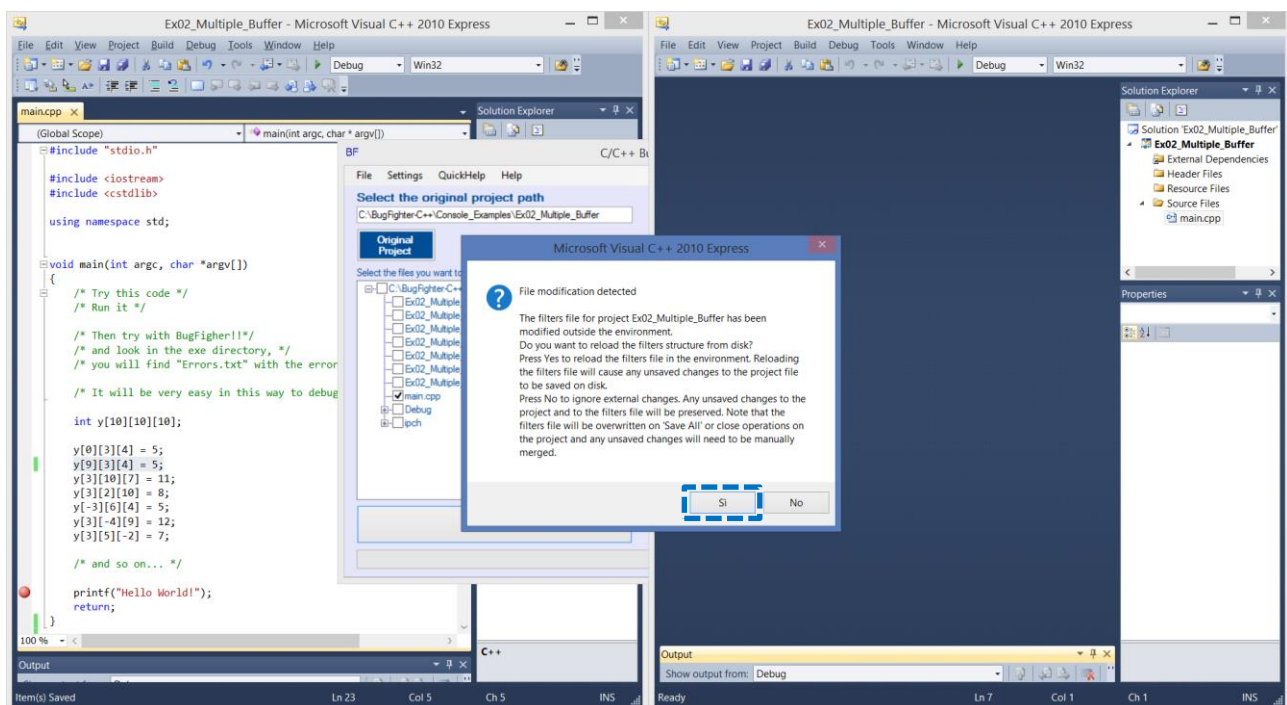
The Properties window on the right shows the properties for the `main` function, including 'Name', 'File', 'FullName', 'IsInjected', 'IsInline', 'IsOverloaded', 'IsSealed', 'IsTemplate', and 'TypeString'.

The Output window at the bottom is empty.

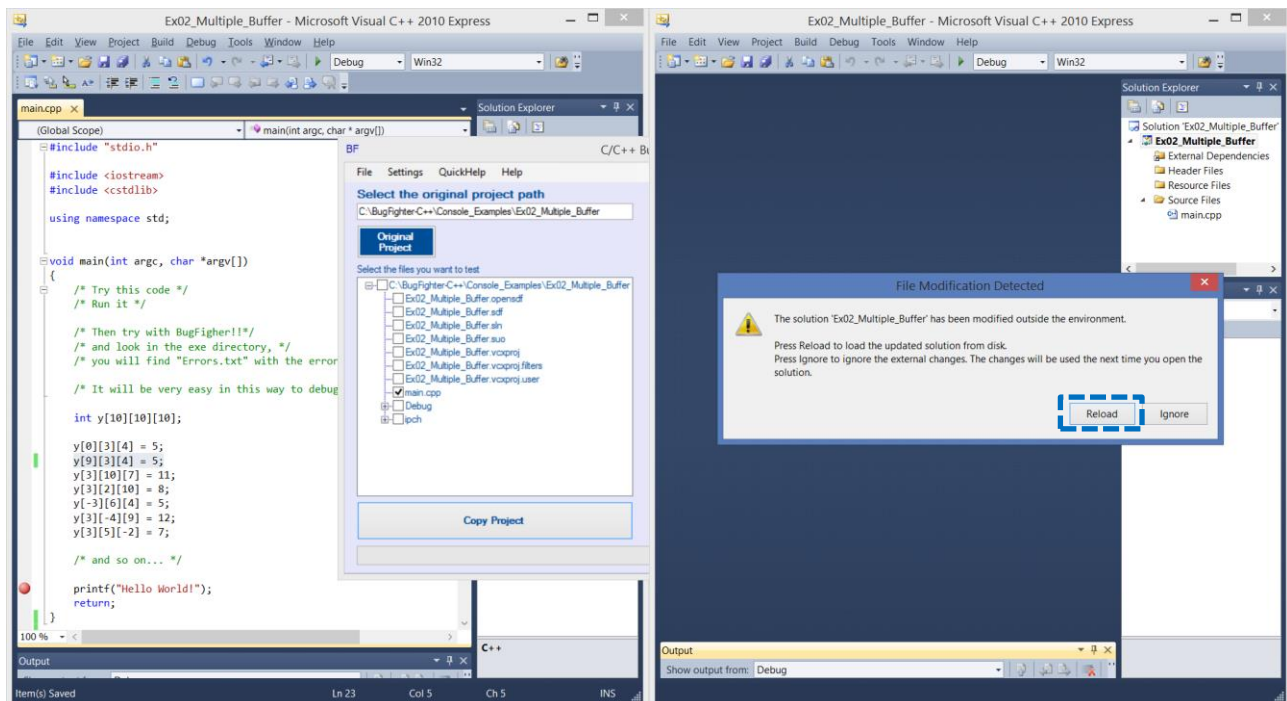
16. Select “Copy Project”



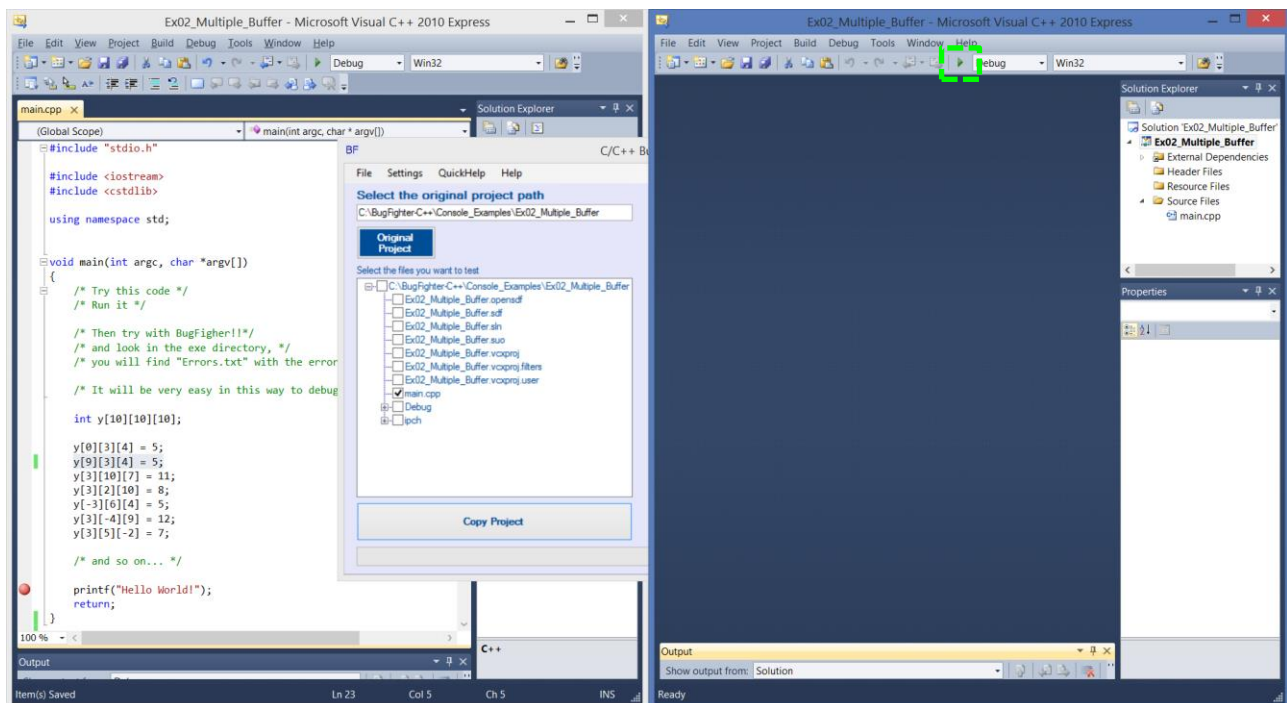
17. Select “Yes” to reload the project



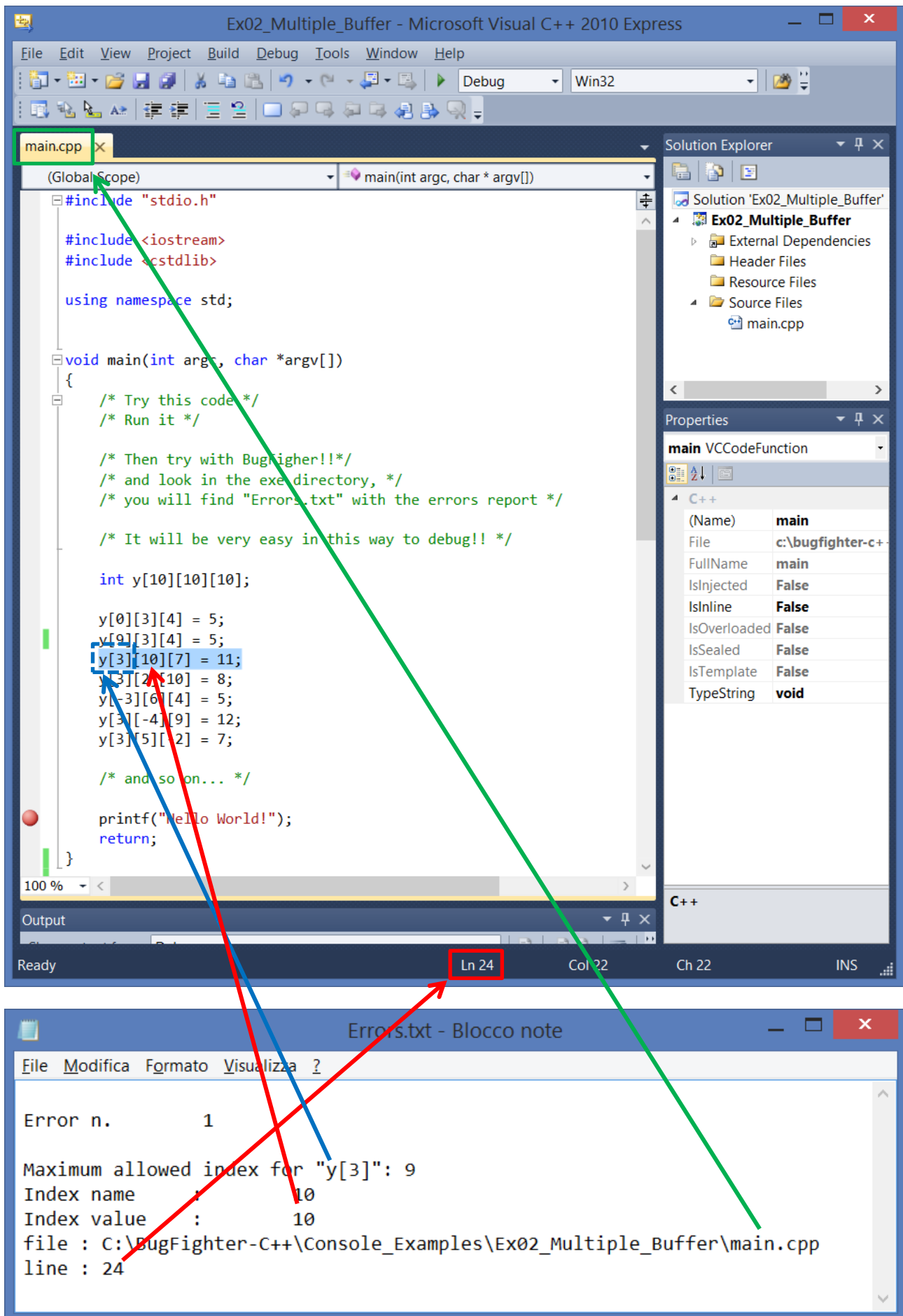
18. Select "Reload"



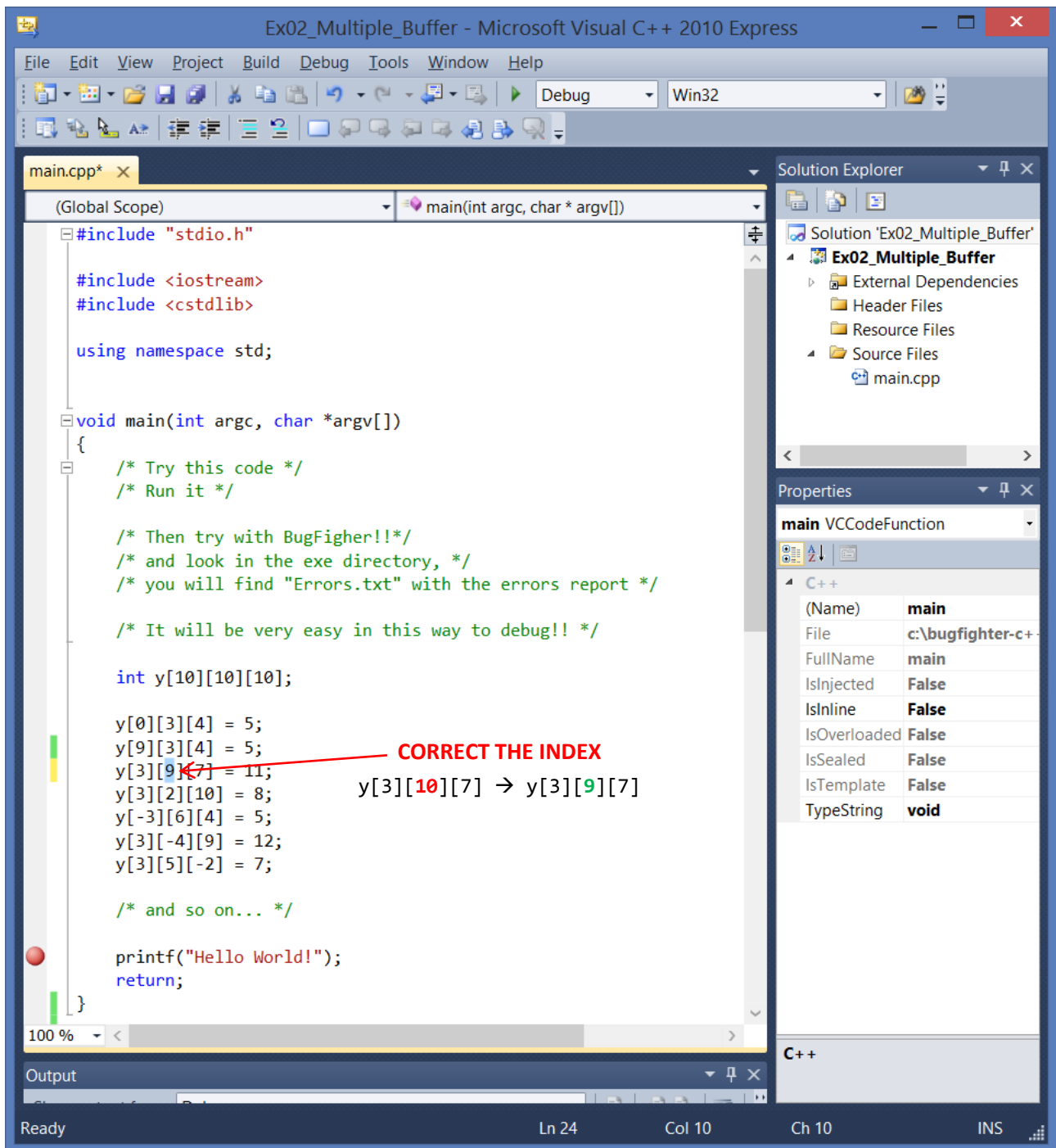
19. Select "Run"



20. Check "Errors.txt"

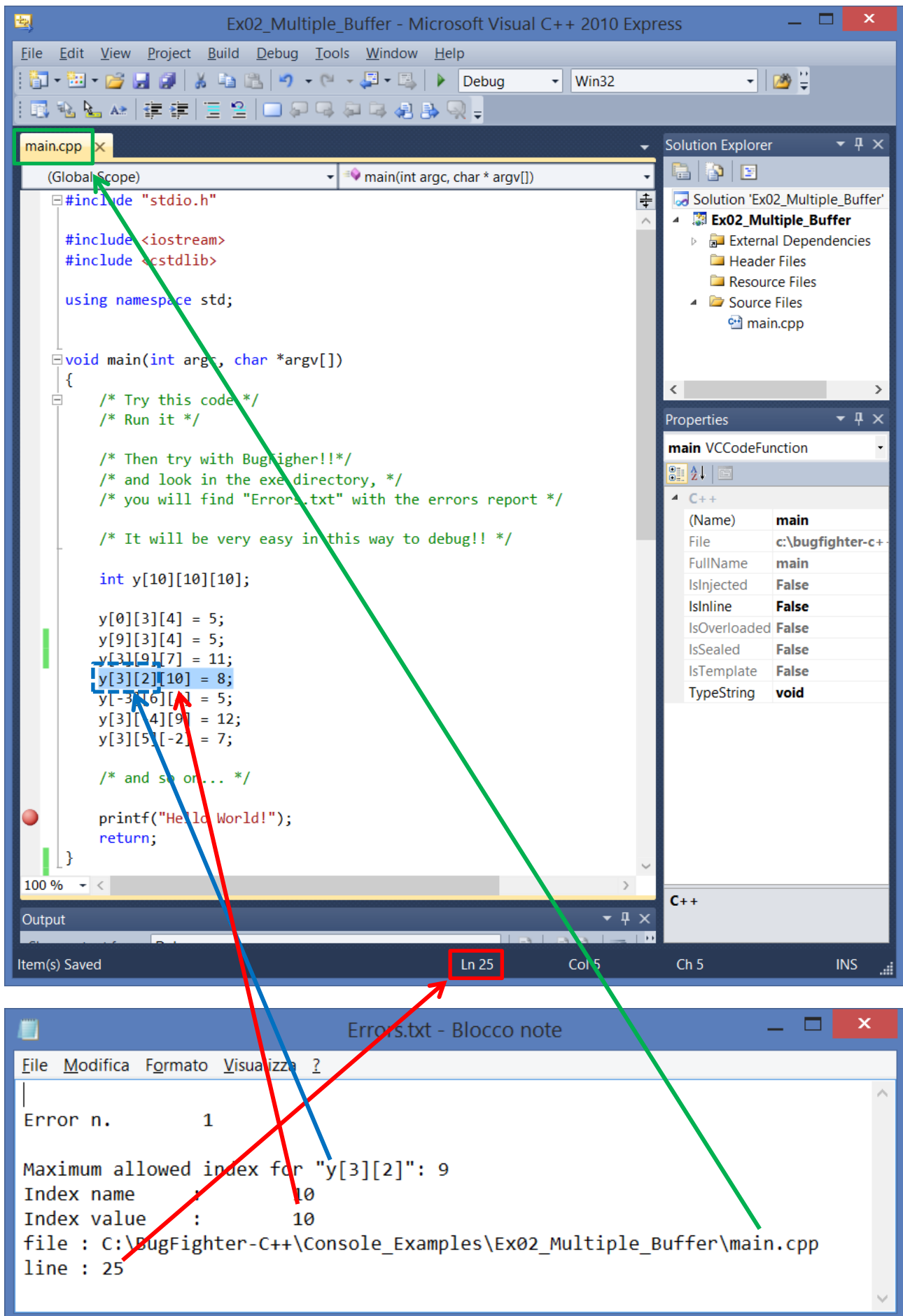


21. Correct the index

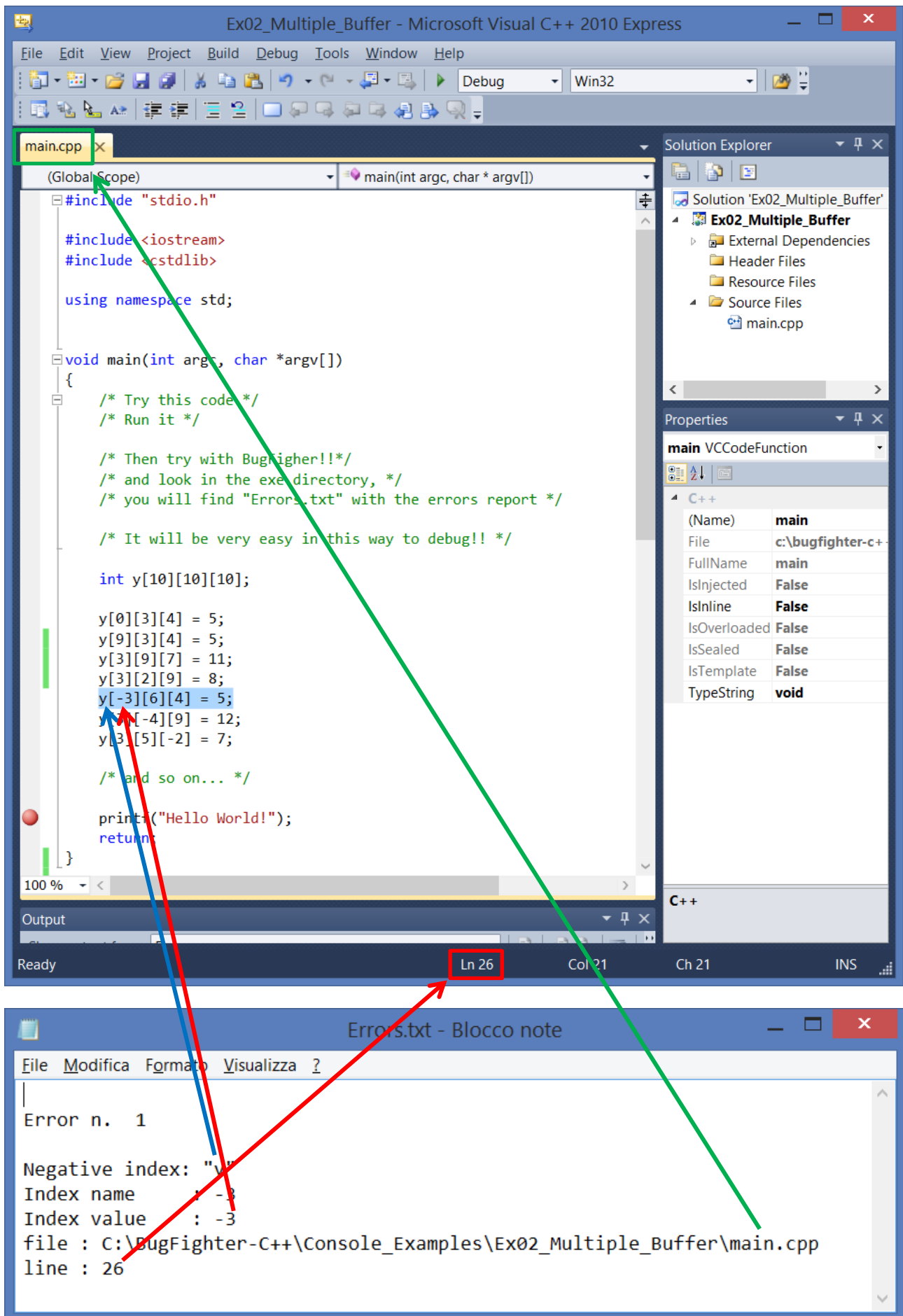


22. Repeat from step 16

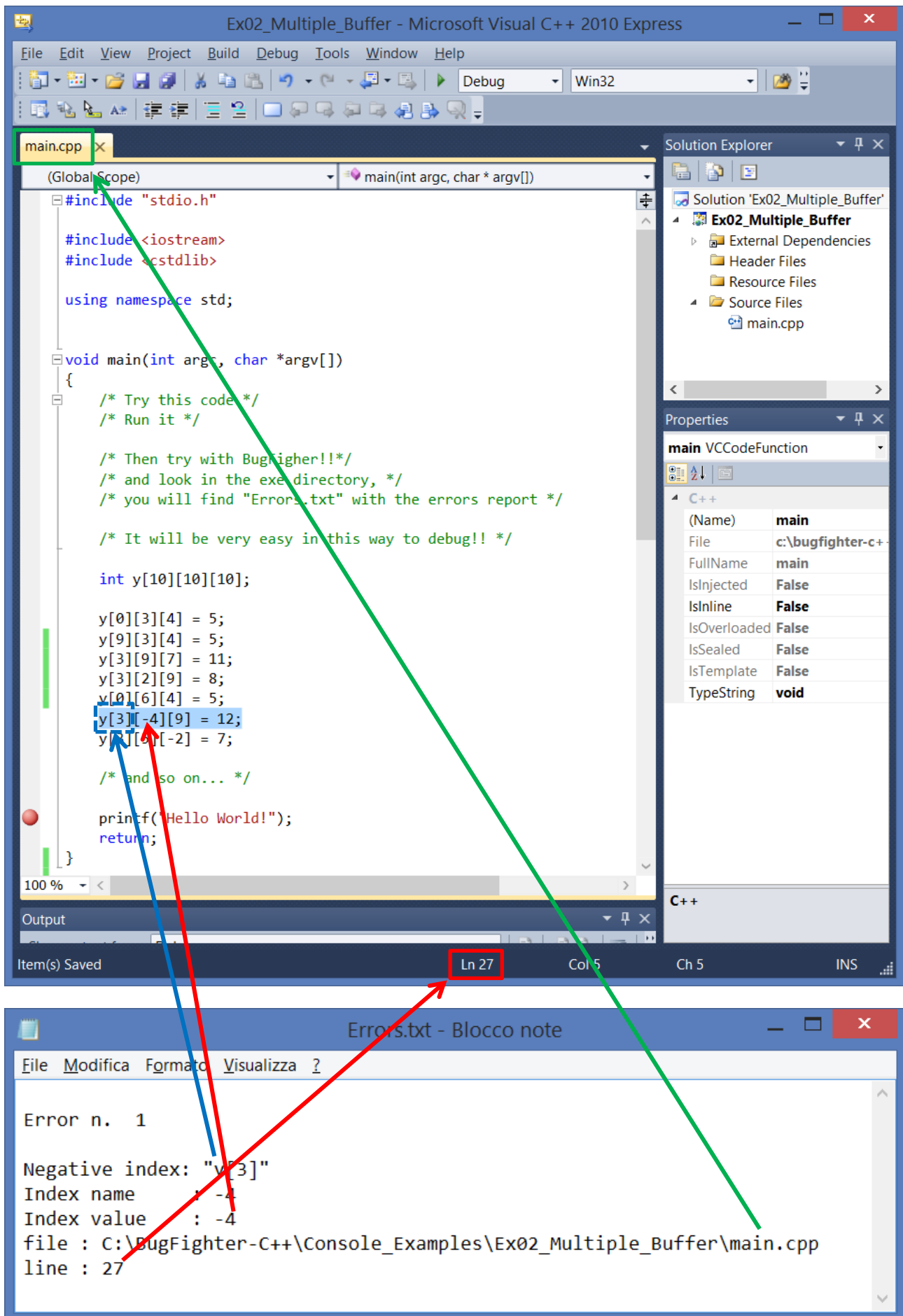
23. Check "Errors.txt"



24. Check "Errors.txt"



25. Check "Errors.txt"



26. Check "Errors.txt"

