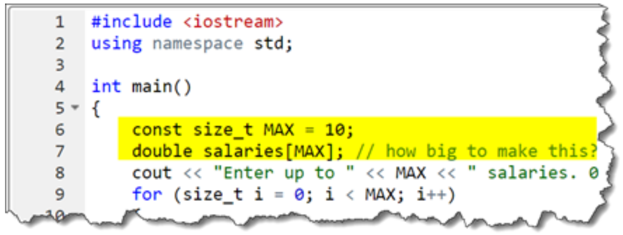
An Array Example

Take a look at this example. How many salaries do I need? Here I've planned on `10`, but if I **need** more, there is no `push_back()`, as there is with `vector`, which would allow expansion. This program is **limited** to a maximum of `10` salaries.



```
#include <iostream>
using namespace std;

int main()
{
    const size_t MAX = 10;
    double salaries[MAX]; // how big to make this?
    cout << "Enter up to " << MAX << " salaries. 0
    for (size_t i = 0; i < MAX; i++)
```

The loop itself can end in one of three ways:

- The user can enter `10` salaries and, because the array is full, the loop will end. **All** of the elements in the array **will be used**.
- The user can enter a `0` as a salary (**the sentinel**), and the loop will terminate. Only **some** of the elements **will be used** in the array.
- The user can enter a non-numeric value such as the word `"quit"` and the `cin` object will enter the **fail state** when trying to read the next salary. The *`if`* statement inside the loop checks for this and exits when this occurs.

Once you exit the loop, you **don't know which** of these occurred. Even worse, you can't tell how many elements are actually valid, and which are unused. Let's fix that.