

Array Intro

```
{  
int main()  
{
```

	itemsAr	index	sum
0	5	✓0	✓5
1	10	✓1	✓15
2	15	✓2	✓30

```
    Const int MAX_AR = 3;
```

```
    int itemsAr[MAX_AR] = {5, 10, 15};
```

```
    int Sum;
```

```
    int index;
```

```
    Sum = 0;
```

```
    for (index = 0; index < MAX_AR; index++)
```

```
    {  
        Cout << "Enter an integer: ";  
        Cin >> itemsAr[index];
```

```
        Sum = Sum + itemsAr[index];  
    }
```

```
    Cout << "The sum of the numbers = " << Sum << endl;  
    Cout << "The numbers in reverse are: ";
```

```
    for (index = MAX_AR - 1; index > -1; index--)
```

```
    {  
        Cout << itemsAr[index] << ", ";  
    }
```

```
    Return 0;
```

15, 10, 5

```
}
```

E27 part 2 1/2

```
int itemsAr[AR_SIZE] = {0};  
// INPUT - read input from a file into the array  
index = 0;
```

```
while (inFile && index < AR_SIZE)  
{  
    inFile >> itemsAr[index];
```

```
    index++;
```

```
}
```

```
// SEARCH - for searchItem in the array
```

```
searchItem = 10;
```

```
index = 0;
```

```
found = false;
```

0, 1, 2

6, 6, 6

F, F, T

```
while (index < AR_SIZE && !found)
```

```
{
```

```
    if (itemsAr[index] == searchItem)
```

```
    {
```

```
        found = true;
```

```
    }
```

```
else
```

```
{
```

```
    index++;
```

```
}
```

```
}
```

itemsAr

0	1	2	3	4	5
3	7	10			

INPUT FILE

3 7 10 2 11

AR_SIZE index found searchItem

6

0

false

10

6

1

false

10

6

2

true

10

E27 part 2 [2/2]

Const int AR_SIZE = 6;

int itemsAr[AR_SIZE] = {3, 7, 10, 2, 10, 12};

int index;

int searchItem;

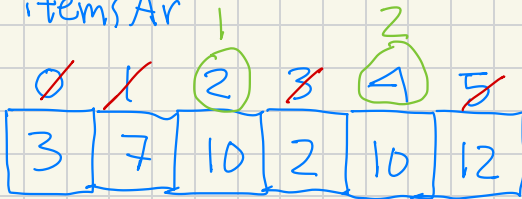
int instanceCount;

searchItem = 10;

instanceCount = 0;

```
for(index = 0; index < AR_SIZE; index++)  
{  
    if(itemsAr[index] == searchItem)  
    {  
        ++instanceCount;  
    }  
}
```

itemsAr



<u>AR_SIZE</u>	<u>index</u>	<u>instanceCount</u>	<u>searchItem</u>
6	0	0	10
6	1	0	10
6	2	1	10
6	3	1	10
6	4	2	10
6	5	2	10

Looping in Arrays

How do we know when to use a For loop vs a While loop?

If we need to check every element of our array, we use a for loop.

If we need to check our array for 1 specific value, we use a while loop to search for it until it is found once.

What is the one check we have in every loop when using arrays?



So if I am calculating the average for the array do I have to inspect every element? Yes, because you need to add every index value into a total in order to divide by the count (using a for loop.)

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12
13 #include <iostream>
14 #include <iomanip>
15 #include <string>
16 #include <fstream>
17 #include <sstream>
18 using namespace std;
19
20
21
22 void ReadArrayFromFile (int numberArray[], const int AR_SIZE);
23
24 string ArrayToStringOss(int numberArray[], const int AR_SIZE);
25
26 void OutputArrayToFile (int numberArray[], const int AR_SIZE);
27
28 double AverageOfArray (int numberArray[], const int AR_SIZE);
29
30
31
32 #endif /* HEADER_H_ */
33
```

E27_Input.txt

Sunday, November 24, 2024, 7:44 PM

1 18 25 32 18 16 54 25 37 18 64

E27_Output.txt

Sunday, November 24, 2024, 7:44 PM

1 18 25 32 18 16 54 25 37 18 64

```
1 Enter an integer: 5
2 Enter an integer: 10
3 Enter an integer: 15
4 The sum of the numbers is = 30
5 The numbers in reverse are: 15, 10, 5,
```



```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 int main()
14 {
15     //CONSTANT
16     const int AR_SIZE = 10;
17
18     //VARIABLES
19     int numArray[AR_SIZE] = {0};
20     int index;
21     string arrayStringIn;
22     string arrayStringOut;
23     double arrayAvg;
24
25
26     //PROCESSING - calling functions
27     ReadArrayFromFile(numArray, AR_SIZE);
28
29     arrayStringIn = ArrayToStringOss(numArray, AR_SIZE);
30
31     arrayAvg = AverageOfArray(numArray, AR_SIZE);
32
33
34     //OUTPUT - output converted string to console & file
35     cout << arrayStringIn;
36
37     OutputArrayToFile(numArray, AR_SIZE);
38
39     index = FindFirstInstance(numArray, AR_SIZE, 10);
40
41     if (index != -1)
42     {
43         cout << "\nValue 10 found at index: " << index << endl;
44     }
45     else
46     {
47         cout << "\nValue 10 not found! \n";
48     }
49
50
51
```

```
52     cout << setprecision(3);
53     cout << "\n\nThe average of the provided array is: " << arrayAvg << endl;
54     cout << setprecision(6);
55
56
57
58
59     return 0;
60 }
61
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 void ReadArrayFromFile(int numArray[], const int AR_SIZE)
14 {
15     ifstream inFile;
16     int index;
17
18     inFile.open("E27_Input.txt");
19
20     //load the array from a file
21     index = 0;
22
23
24     while (inFile && index < AR_SIZE)
25     {
26         inFile >> numArray[index];
27
28         index++;
29     }
30
31     inFile.close();
32
33 }
34
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #include "header.h"
11
12 string ArrayToStringOss(int numArray[], const int AR_SIZE)
13 {
14     ostringstream outOss;
15     int          index;
16
17     index = 0;
18
19     while (index < AR_SIZE)
20     {
21         outOss << numArray[index] << " ";
22
23         index++;
24     }
25
26     return outOss.str();
27
28 }
29
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 void OutputArrayToFile(int numArray[], const int AR_SIZE)
14 {
15     ofstream outFile;
16     int index;
17
18     outFile.open("E27_Output.txt");
19
20     index = 0;
21
22     while (outFile && index < AR_SIZE)
23     {
24         outFile << numArray[index] << " ";
25
26         index++;
27     }
28
29     outFile.close();
30 }
31
32
33
34
35
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 double AverageOfArray(int numArray[], const int AR_SIZE)
14 {
15     int    index;
16     double arraySum;
17
18     arraySum = 0.0;
19     index    = 0;
20
21     while (index < AR_SIZE)
22     {
23         arraySum = arraySum + numArray[index];
24
25         index++;
26     }
27
28     return arraySum / AR_SIZE;
29 }
30
31
32
33
34
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12
13 #include <iostream>
14 #include <iomanip>
15 #include <string>
16 #include <fstream>
17 #include <sstream>
18 using namespace std;
19
20
21 void ReadArrayFromFile (int numberArray[], const int AR_SIZE);
22
23 string ArrayToStringOss(int numberArray[], const int AR_SIZE);
24
25 void OutputArrayToFile (int numberArray[], const int AR_SIZE);
26
27 double AverageOfArray (int numberArray[], const int AR_SIZE);
28
29 int FindFirstInstance (int numberArray[], const int AR_SIZE,
30                       int &elementValue);
31
32 int FindAllInstances (int numberArray[], const int AR_SIZE,
33                      int &elementValue);
34
35 #endif /* HEADER_H_ */
36
```

E27_Input.txt

Sunday, November 24, 2024, 10:38 PM

1 18 25 32 18 16 54 25 37 18 64

E27_Output.txt

Sunday, November 24, 2024, 10:33 PM

1 18 25 32 18 16 54 25 37 18 64

```
1 18 25 32 18 16 54 25 37 18 64
2
3 The average of the provided array is: 30.7
4
5 Value 32 located at index: 2
6
7 18 25 32 18 16 54 25 37 18 64
8
9 The average of the provided array is: 30.7
10
11 Value 64 located at index: 9
12
13 18 25 32 18 16 54 25 37 18 64
14
15 The average of the provided array is: 30.7
16
17 Value 16 located at index: 4
18
19 18 25 32 18 16 54 25 37 18 64
20
21 The average of the provided array is: 30.7
22
23 Value 2000 not found!
24
25
```

```
1 18 25 32 18 16 54 25 37 18 64
2
3 The average of the provided array is: 30.7
4
5 Value 18 located at index: 0
6
7 Value found at index/indices: 0 3 8
8
9 18 25 32 18 16 54 25 37 18 64
10
11 The average of the provided array is: 30.7
12
13 Value 64 located at index: 9
14
15 Value found at index/indices: 9
16
17 18 25 32 18 16 54 25 37 18 64
18
19 The average of the provided array is: 30.7
20
21 Value 25 located at index: 1
22
23 Value found at index/indices: 1 6
24
25 18 25 32 18 16 54 25 37 18 64
26
27 The average of the provided array is: 30.7
28
29 Value 25 located at index: 1
30
31 25 was found at index/indices: 1 6
32
33 18 25 32 18 16 54 25 37 18 64
34
35 The average of the provided array is: 30.7
36
37 Value 8000000 not found!
38
39
40
41
42
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 int main()
14 {
15     //CONSTANT
16     const int AR_SIZE = 10;
17
18     //VARIABLES
19     int numArray[AR_SIZE] = {0};
20     int index;
21     string arrayStringIn;
22     string arrayStringOut;
23     double arrayAvg;
24     int elementValue;
25     int allInstances;
26
27     //INITIALIZE
28     elementValue = 8000000;
29
30
31     //PROCESSING - calling functions
32     ReadArrayFromFile(numArray, AR_SIZE);
33
34     arrayStringIn = ArrayToStringOss(numArray, AR_SIZE);
35
36     arrayAvg = AverageOfArray(numArray, AR_SIZE);
37
38     index = FindFirstInstance(numArray, AR_SIZE, elementValue);
39
40
41     //OUTPUT - output converted string to console & file
42     cout << arrayStringIn;
43
44     OutputArrayToFile(numArray, AR_SIZE);
45
46     cout << setprecision(3);
47
48     cout << "\n\nThe average of the provided array is: " << arrayAvg << "\n\n";
49
50     cout << setprecision(6);
51 }
```

```
52
53     if (index != -1)
54     {
55         cout << "Value " << elementValue << " located at index: " << index;
56         cout << "\n\n";
57     }
58     else
59     {
60         cout << "Value " << elementValue << " not found!\n";
61         cout << "\n\n";
62     }
63
64     allInstances = FindAllInstances(numArray, AR_SIZE, elementValue);
65
66     cout << "\n\n";
67
68
69     return 0;
70 }
71
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 void ReadArrayFromFile(int numArray[], const int AR_SIZE)
14 {
15     ifstream inFile;
16     int index;
17
18     inFile.open("E27_Input.txt");
19
20     //load the array from a file
21     index = 0;
22
23
24     while (inFile && index < AR_SIZE)
25     {
26         inFile >> numArray[index];
27
28         index++;
29     }
30
31     inFile.close();
32
33 }
34
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12 string ArrayToStringOss(int numArray[], const int AR_SIZE)
13 {
14     ostringstream outOss;
15     int          index;
16
17     index = 0;
18
19     while (index < AR_SIZE)
20     {
21         outOss << numArray[index] << " ";
22
23         index++;
24     }
25
26     return outOss.str();
27
28 }
29
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 void OutputArrayToFile(int numArray[], const int AR_SIZE)
14 {
15     ofstream outFile;
16     int index;
17
18     outFile.open("E27_Output.txt");
19
20     index = 0;
21
22     while (outFile && index < AR_SIZE)
23     {
24         outFile << numArray[index] << " ";
25
26         index++;
27     }
28
29     outFile.close();
30 }
31
32
33
34
35
```



```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12
13 double AverageOfArray(int numArray[], const int AR_SIZE)
14 {
15     int    index;
16     double arraySum;
17
18     arraySum = 0.0;
19     index    = 0;
20
21     while (index < AR_SIZE)
22     {
23         arraySum = arraySum + numArray[index];
24
25         index++;
26     }
27
28     return arraySum / AR_SIZE;
29 }
30
31
32
33
34
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27    : Arrays 4/4
5  * CLASS      : CS1A
6  * SECTION    : MW 8am
7  * DUE DATE   : 11/18/24
8  *****/
9
10 #include "header.h"
11
12 int FindFirstInstance(int numArray[], const int AR_SIZE, int &elementValue)
13 {
14     int index;
15     int result;
16
17     result = -1;
18
19
20     for (index = 0; index < AR_SIZE; index++)
21     {
22         if (numArray[index] == elementValue && result == -1)
23         {
24             result = index;
25         }
26     }
27
28
29     return result;
30 }
31
32
```

```
1 /*****
2  * AUTHOR      : Blake Allard
3  * STUDENT ID  : 358888
4  * EXR #27     : Arrays 4/4
5  * CLASS       : CS1A
6  * SECTION     : MW 8am
7  * DUE DATE    : 11/18/24
8  *****/
9
10 #include "header.h"
11
12 int FindAllInstances(int numArray[], const int AR_SIZE, int &elementValue)
13 {
14     int index;
15     int count;
16
17     count = 0;
18
19
20     for (index = 0; index < AR_SIZE; index++)
21     {
22
23         if (numArray[index] == elementValue)
24         {
25             count++;
26         }
27     }
28
29     if (count > 0)
30     {
31         cout << elementValue << " was found at index/indices: ";
32
33         for (index = 0; index < AR_SIZE; index++)
34         {
35
36             if (numArray[index] == elementValue)
37             {
38                 cout << index << " ";
39             }
40         }
41     }
42
43     return count;
44 }
45
```