

CSCI6033_Lab2_Yibo_Zhu

Constructor-Array (int size, string input)

Precondition:

Size is a “int” type, bigger than 0

Input is a string of comma-separated characters

Postcondition:

Input string is stored in a char array in order.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
-10
Please enter a series of comma-separated characters for your array
a,s,f,g
a.out: Yibo_Zhu_lab2.cpp:25: Array::Array(int&, std::__cxx11::string
&): Assertion `size > 0' failed.
Aborted (core dumped)
```

CWE-20: Improper Input Validation

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
-10
Please enter a series of comma-separated characters for your array
a,b,c
Size should be bigger than 0!
Please enter size again:
10
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
4
a b c
```

I use improper input size = -10 to test the precondition. To handle this error, I use a while loop to request a new input for size, if the size still is less than 0, repeat the request.

Destructor- ~Array()

Precondition:

Char* array = new char[size] (new memory is allocated).

Postcondition:

Deallocate the memory.

Char ReadFromArray (int index)

Precondition:

Index is valid ($0 \leq \text{index} < \text{size}$)

Postcondition:

Return the character at the specific index

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,g,h,e,t,y
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
1
Enter an index to read a value from the array
-1
a.out: Yibo_Zhu_lab2.cpp:78: char Array::ReadFromArray(int&): Assertion `index >= 0 && index < size' failed.
Aborted (core dumped)
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,g,d,h,e,t
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
1
Enter an index to read a value from the array
100
a.out: Yibo_Zhu_lab2.cpp:78: char Array::ReadFromArray(int&): Assertion `index >= 0 && index < size' failed.
Aborted (core dumped)
```

CWE-125: Out-of-bounds Read

I use out-of-bound index and try to read the array use that index. To handle the error, I use a while loop(index < 0 || index >= size), if index is not fit the condition, request for a new input.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
d,f,d,f,g,h,j,e,r
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
1
Enter an index to read a value from the array
-1
The valid index is 0 to 8!
Please enter the index again:100
The valid index is 0 to 8!
Please enter the index again:1
The item in index[1] is f
```

Void WriteToArray (int index, char replace)

Precondition:

Index is valid ($0 \leq \text{index} < \text{size}$)

Char replace is valid.

Postcondition:

The character at specific index is replaced by char replace.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,d,f,g,h,r,t,y,u
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
2
Enter an index to write a value to the array
100
What single character would you like to write to the array?
j
a.out: Yibo_Zhu_lab2.cpp:97: void Array::WriteToArray(int&, char&):
Assertion `index >= 0 && index < size' failed.
Aborted (core dumped)
```

CWE-125: Out-of-bounds Read

I use out-of-bound index and try to overwrite a new character into the array. To handle the error, I use a while loop ($\text{index} < 0 \text{ || } \text{index} \geq \text{size}$), if index is not fit the condition, request for a new input.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,g,d,g,d,f,e,r,t,w,e
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
2
Enter an index to write a value to the array
100
What single character would you like to write to the array?
j
The valid index is 0 to 9!
Please enter the index again:1
The item in index[1] is j
```

Void DeleteArray()

Precondition:

Char* array = new char[size] (new memory is allocated).

Postcondition:

Deallocate the memory.

Set the array to NULL.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
100
Please enter a series of comma-separated characters for your array
a,g,b,f,g,e,r,t
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
a.out: Yibo_Zhu_lab2.cpp:115: void Array::DeleteArray(): Assertion
array != NULL' failed.
Aborted (core dumped)
```

Repeat delete behavior won't trigger an error, but it is an invalid behavior. I test it by selecting choice 3 repeatedly. To handle the invalid behavior, I use an if-else statement, if `char* array = NULL`, show the alert info then stop the behavior.

```
yibo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
yibo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
100
Please enter a series of comma-separated characters for your array
a,f,s,g,s,h
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
The array is already deleted!
```

Void PrintArray()

Precondition:

Array is created and has characters stored in it.

Postcondition:

Display the content of the array.

```
ytbo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
ytbo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,g,b,d,h
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
4
a.out: Yibo_Zhu_lab2.cpp:132: void Array::PrintArray(): Assertion `array != NULL' failed.
Aborted (core dumped)
```

CWE-416: Use After Free

I delete the array first then try to print the array. To handle the error, I use a if-else statement, if char* array == NULL, redirect to the menu and suggest the user select choice 5.

```
ytbo@yibo-VirtualBox:~/Downloads$ g++ -DASSERT_CHECK main.cpp
ytbo@yibo-VirtualBox:~/Downloads$ ./a.out
Welcome, please enter a maximum size for your array
10
Please enter a series of comma-separated characters for your array
a,d,f,g
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
3
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
4
The array is not initialized yet
Please choose choice 5 to initialize the array first
```

Void NewArray (int size, string input)

Precondition:

Size is a “int” type, bigger than 0

Input is a string of comma-separated characters

Postcondition:

Input string is stored in a char array in order.

```
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
5
Welcome, please enter a maximum size for your array
-10
Please enter a series of comma-separated characters for your array
d,f
a.out: Yibo_Zhu_lab2.cpp:152: void Array::NewArray(int, std::__cxx11::string): Assertion `size > 0' failed.
Aborted (core dumped)
```

CWE-20: Improper Input Validation

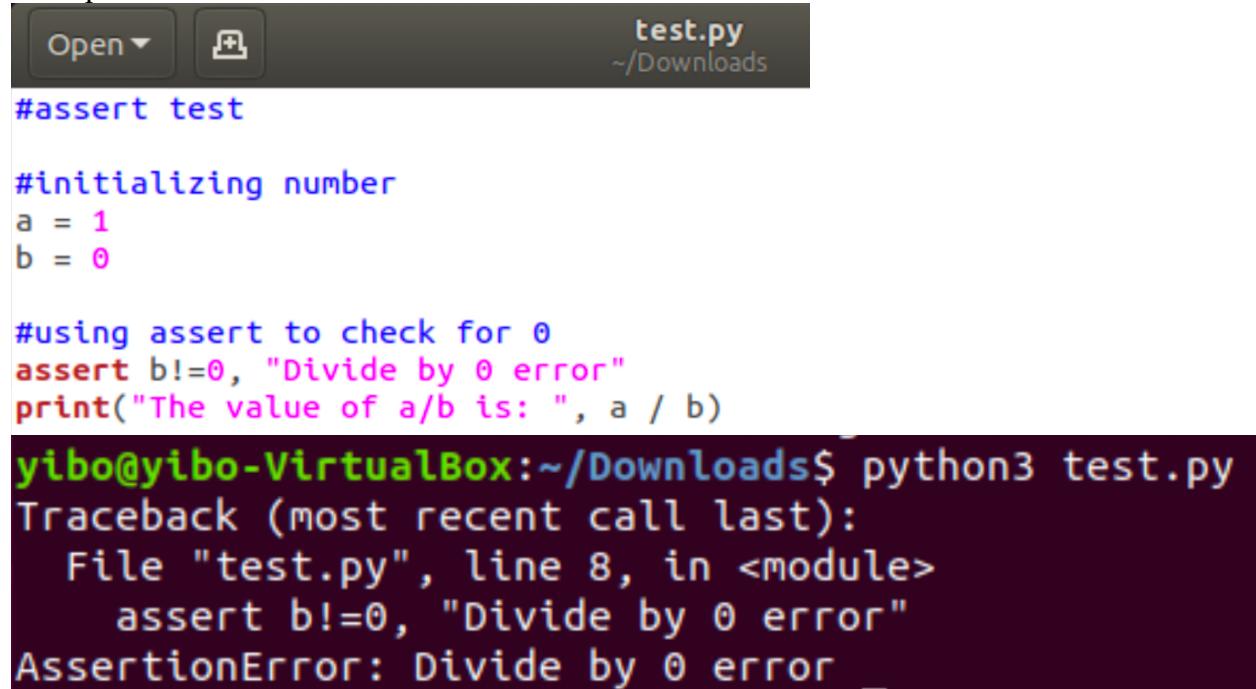
I use improper input size = -10 to test the precondition. To handle this error, I use a while loop to request a new input for size, if the size still is less than 0, repeat it.

```
Array Menu
1. Read by index
2. Write by index
3. Delete array
4. Print array
5. New Array
6. Exit
5
Welcome, please enter a maximum size for your array
-10
Please enter a series of comma-separated characters for your array
a,f,g
Size should be bigger than 0!
Please enter size again:
100
```

Graduate students question:

Yes, python assert keyword helps in smooth flow of code. Like C++, it takes a Boolean condition, which when returns true keep running the program, but if it is computed to be false, it raises an AssertionError along with the optional message provided.

Example:



The screenshot shows a terminal window with the following content:

```
test.py
~/Downloads
```

Open ▾

```
#assert test

#initializing number
a = 1
b = 0

#using assert to check for 0
assert b!=0, "Divide by 0 error"
print("The value of a/b is: ", a / b)
```

```
yibo@yibo-VirtualBox:~/Downloads$ python3 test.py
Traceback (most recent call last):
  File "test.py", line 8, in <module>
    assert b!=0, "Divide by 0 error"
AssertionError: Divide by 0 error
```

Reference:

<https://www.geeksforgeeks.org/python-assert-keyword/>