

Michael Nowak

Texas A&M University

# Overview

MyArray

Removing an item from MyArray

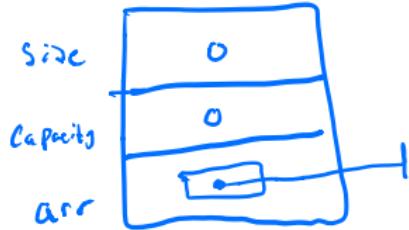
# Overview

MyArray

Removing an item from MyArray

## MyArray

```
1 #ifndef MYARRAY_H
2 #define MYARRAY_H
3 struct MyArray {
4     int *arr = nullptr;
5     int capacity = 0; // no elements can store
6     int size = 0; // no elements currently held
7 };
8 #endif
```



# Overview

MyArray

Removing an item from MyArray

## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```

## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```

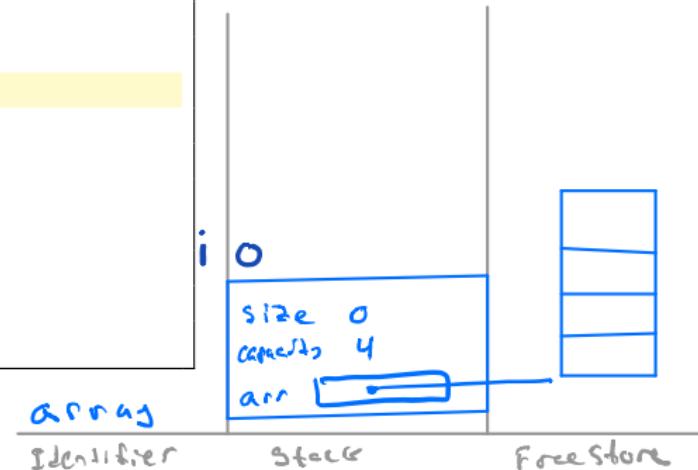
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



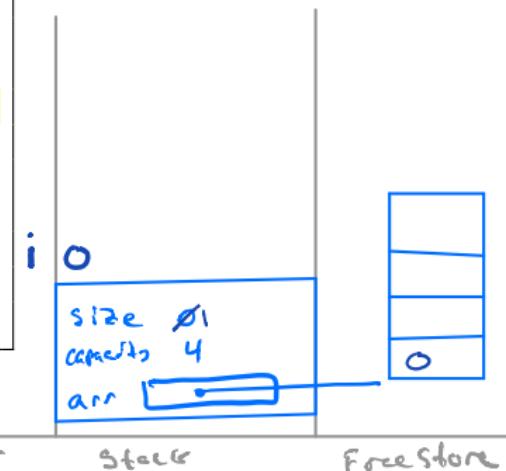
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



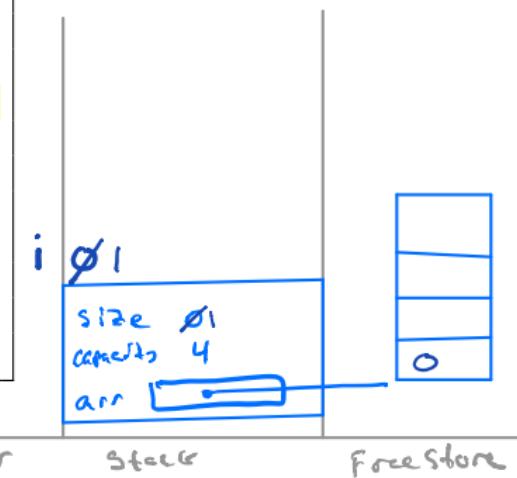
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



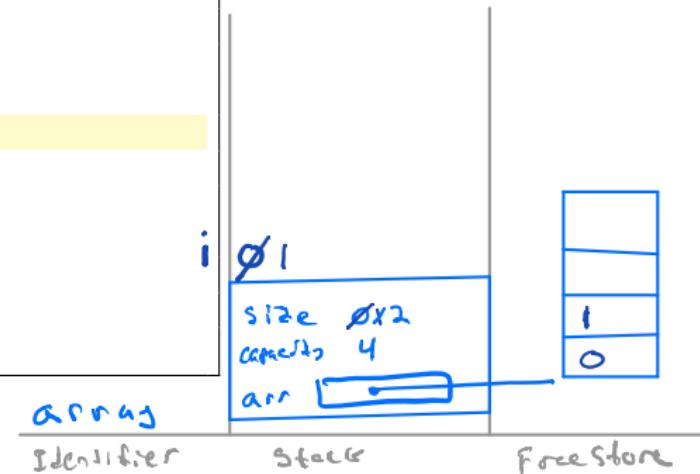
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



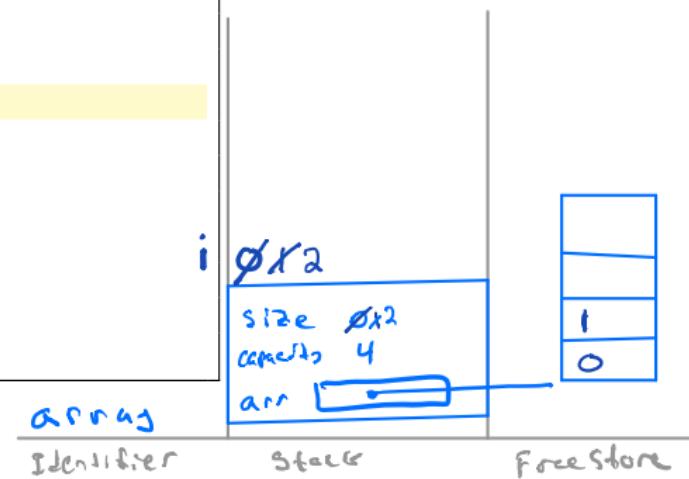
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```

## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



i ~~0x23~~

size ~~0x3~~

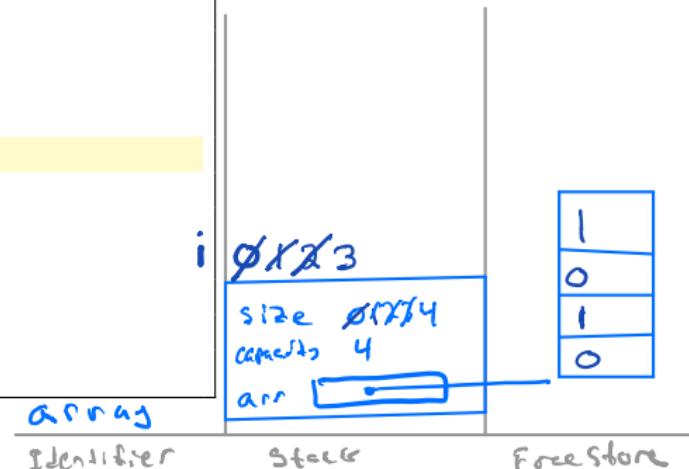
capacity 4

arr [ ]



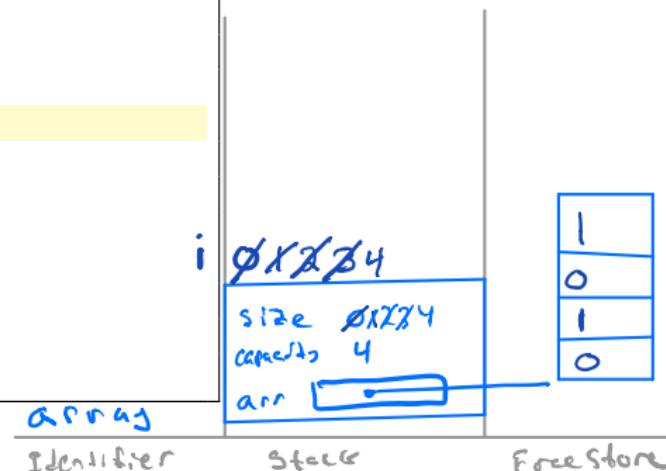
## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



## Removing an item from MyArray

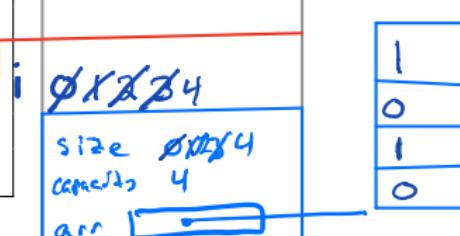
```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



## Removing an item from MyArray

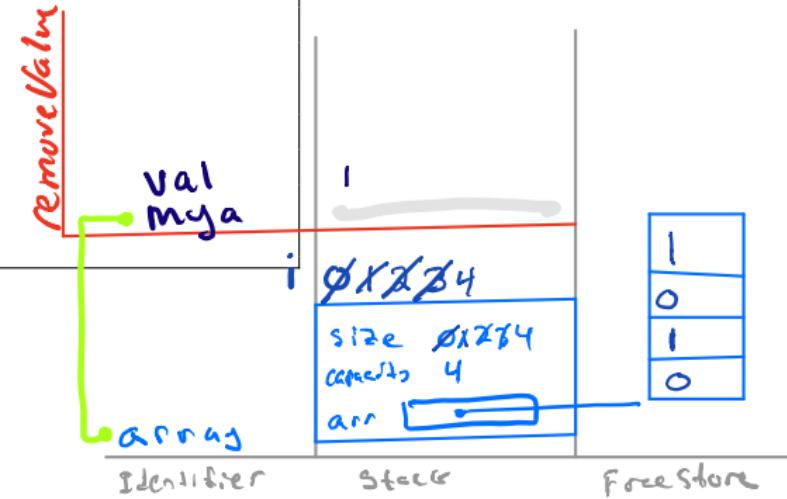
```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```

*removeValue*



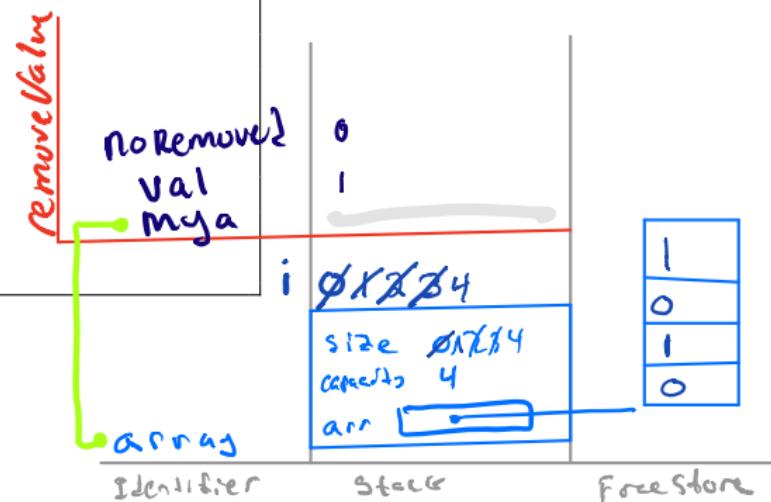
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



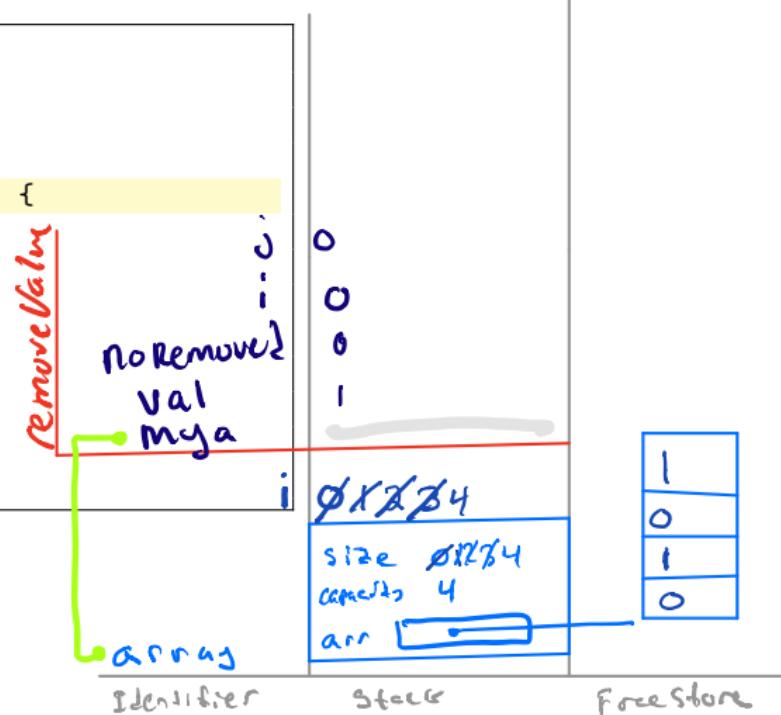
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



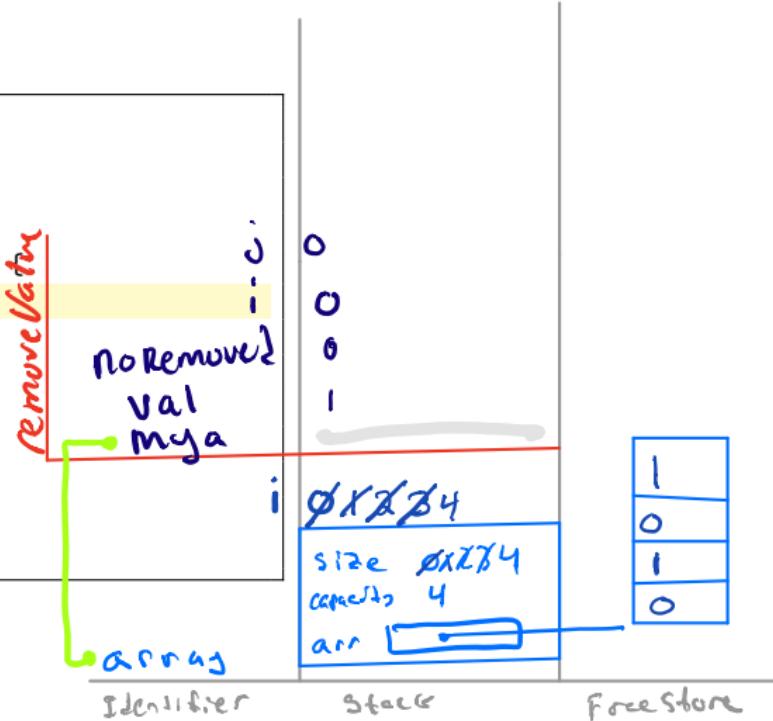
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



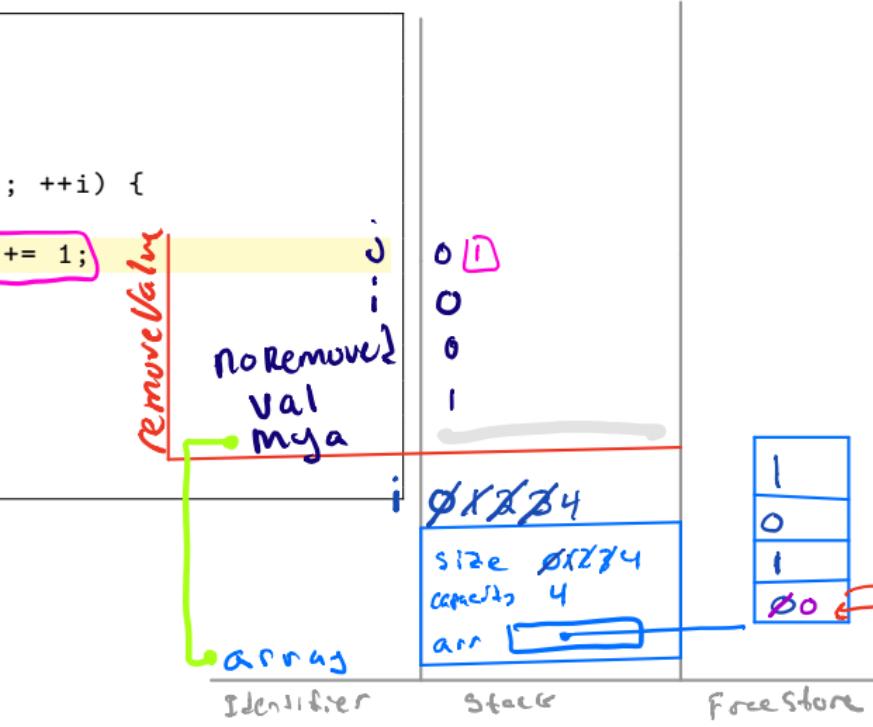
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i)
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else { [o] [o]
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```

removeValue

noRemoved  
val  
mya

array

Identifier

Stack

Stack

j  
i  
0  
1

i

0x284

size 0x284  
capacity 4

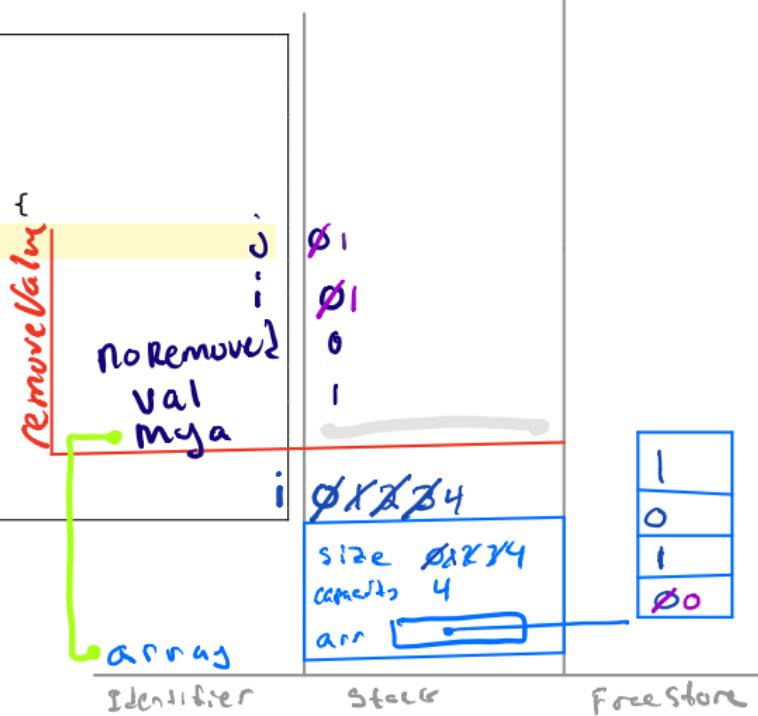
arr [ ]

1
0
1
0

Free Store

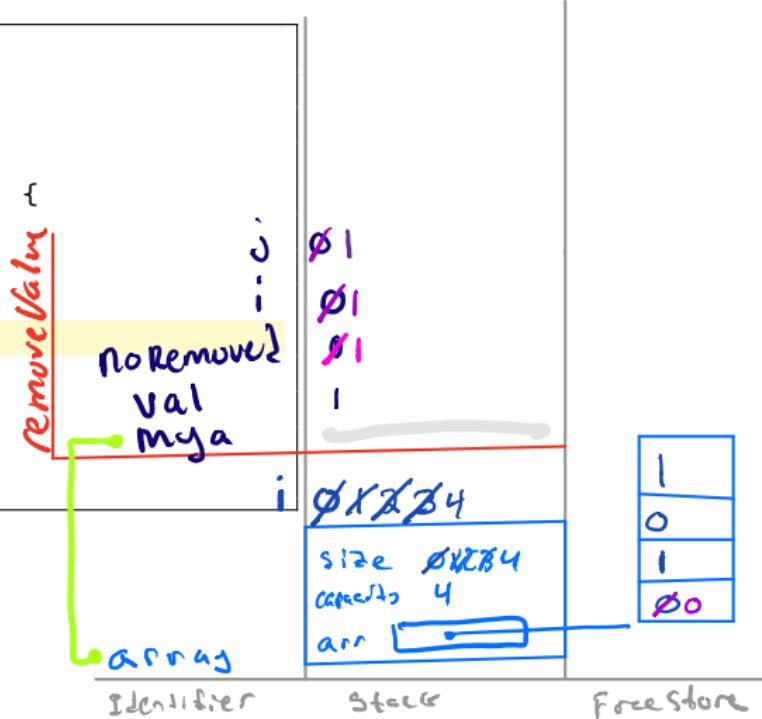
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



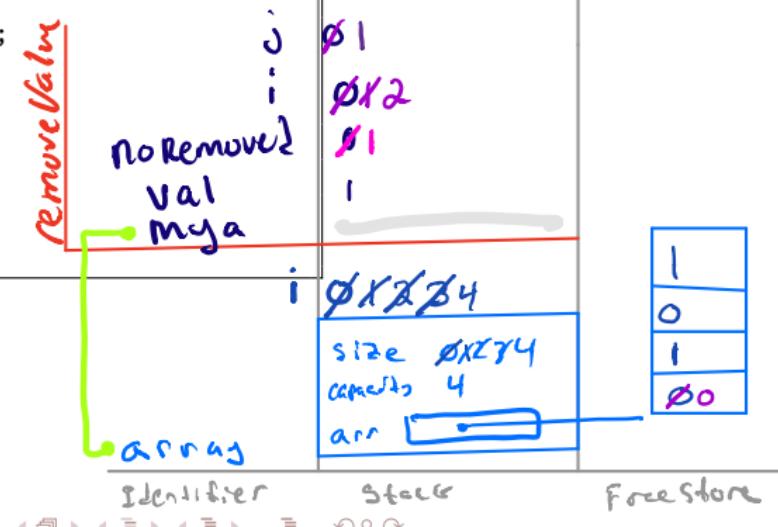
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



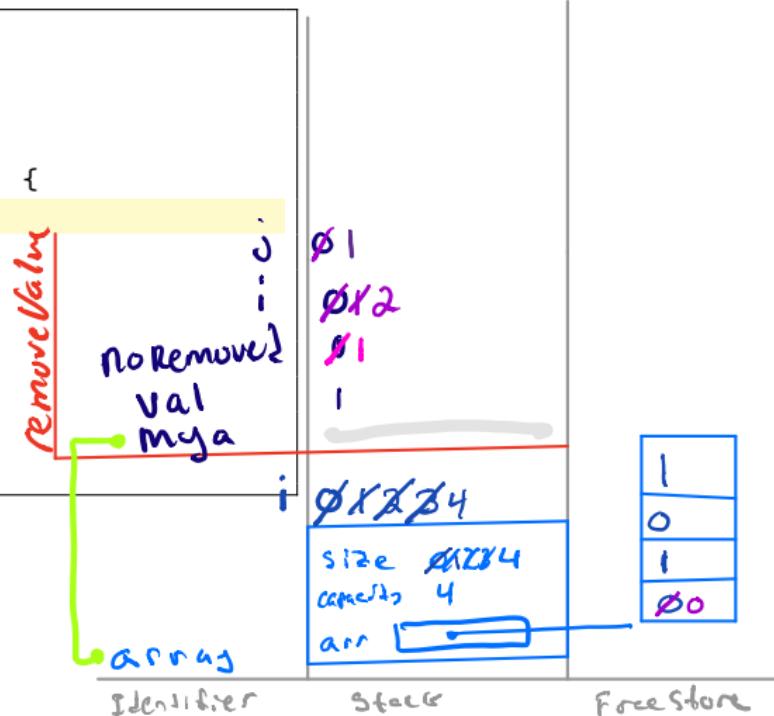
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



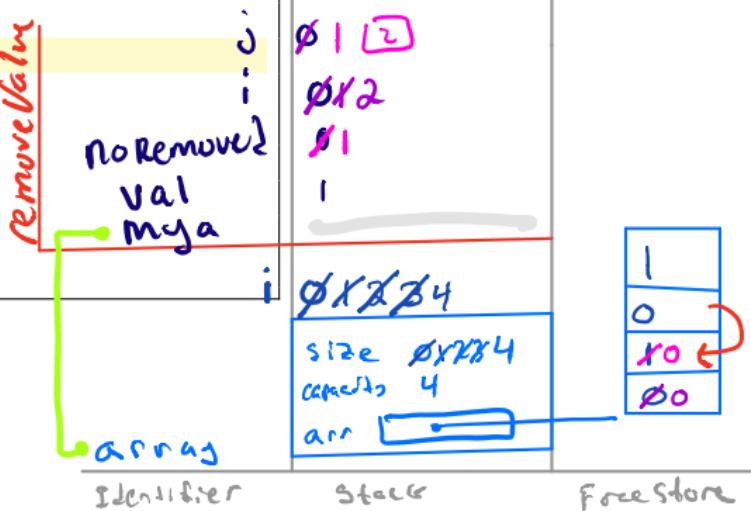
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else { [1] [2]
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```

removeValue

noRemoved  
val  
mya

j

i

!

0x2  
0x23  
#1  
!

i  
0x224

size 0x224  
capacity 4

arr [ ]

1
0
x0
0

## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```

removeValue

noRemoved  
val  
mya

array

j  
0x2  
0x3  
11  
1

i  
0x2b4  
size 0x2b4  
capacity 4  
arr

l  
0  
ro  
po

## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```

removeValue

noRemoved  
val  
mya

array

j  
i  
arr[2]  
arr[3]

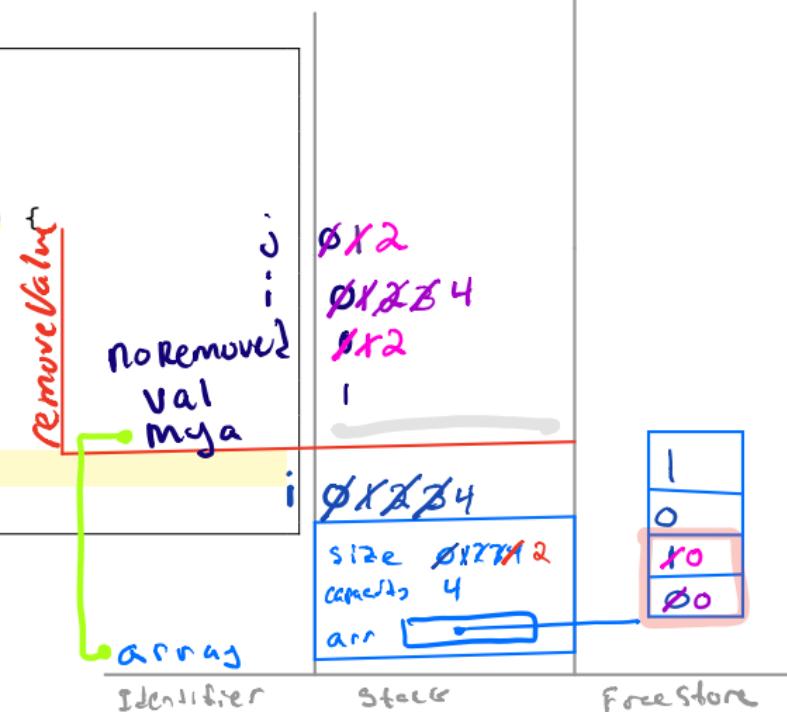
i

arr[4]  
size 4  
current 4  
arr

1
0
x0
00

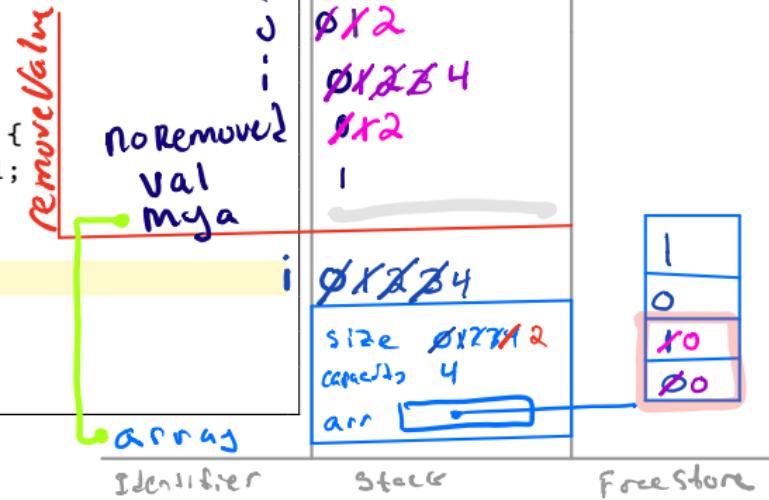
## Removing an item from MyArray

```
1 #include "removeValue.h"
2 void removeValue(MyArray &mya, int val)
3 {
4     int noRemoved = 0;
5     for (int i = 0, j = 0; i < mya.size; ++i) {
6         if (mya.arr[i] != val) {
7             mya.arr[j] = mya.arr[i]; j += 1;
8         } else {
9             noRemoved += 1;
10        }
11    }
12    mya.size -= noRemoved;
13 }
```



## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16
17    return 0;
18 }
```



## Removing an item from MyArray

```
1 #include <iostream>
2 #include "MyArray.h"
3 #include "removeValue.h"
4
5 using namespace std;
6
7 int main()
8 {
9     MyArray array {new int[4], 4, 0};
10
11    for (int i = 0; i < array.capacity; ++i) {
12        array.arr[i] = i % 2; array.size += 1;
13    }
14
15    removeValue(array, 1);
16    delete [] array.arr;
17
18 }
```

removeValue

no removed  
val mya

array

0x2  
0x284  
0x2  
1

i 0x284  
size 0x284  
capacity 4  
arr [ ]

