

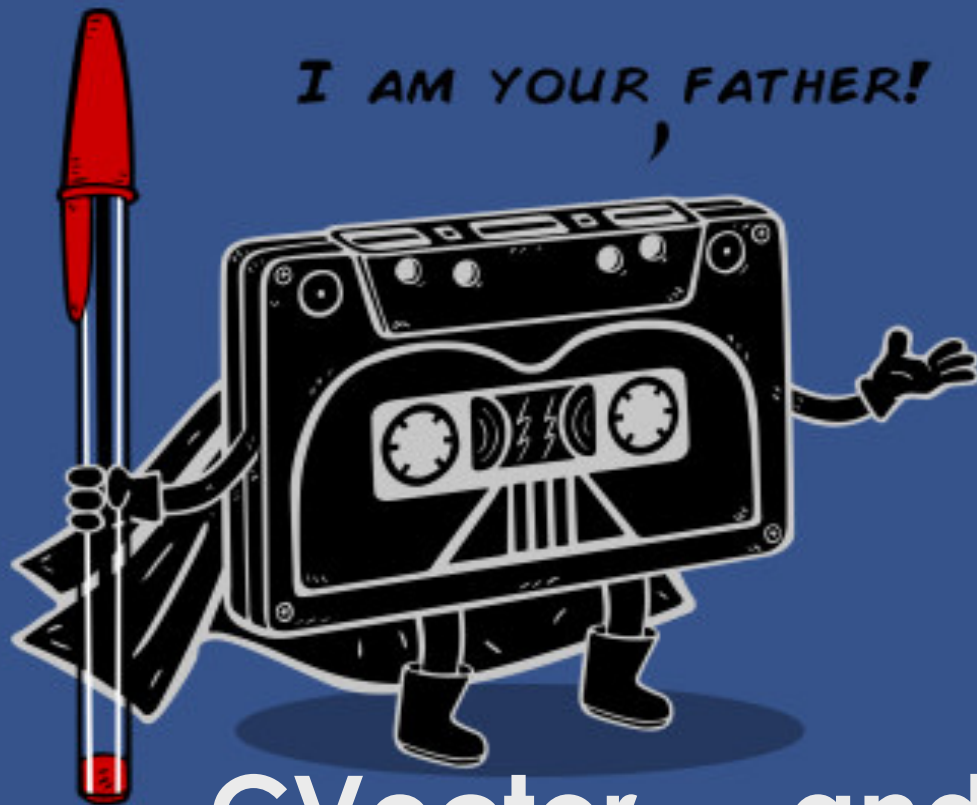
Homework 3

CVector and MyString

2021/4/17 SAT. 12:00 NOON DUE (延長到週六)

2021/4/24 SAT. 23.59 為最後補交期限

**A long time ago in a galaxy far,
far away....**



NOOO!!!



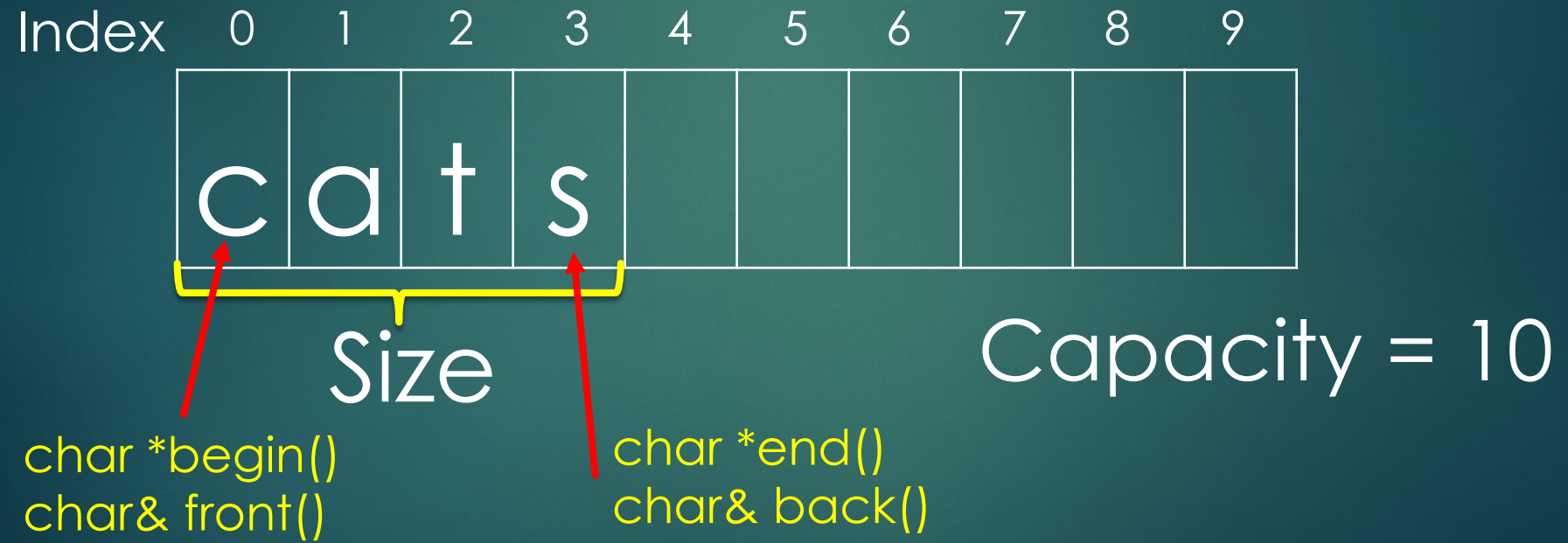
CVector and MyString

What is the CVector?

- ▶ Similar to `std::vector`, the `CVector` provide a flexible container of characters. The necessary functions includes...
 - ▶ Constructors
 - ▶ `front()`, `back()`,
 - ▶ `getSize()`, `resize()`, `getCapacity()`, `reserve()`, `shrink_to_fit()`, `clear()`
 - ▶ `push_back()`, `pop_back()`
 - ▶ `operator[]` for const and non-const object
 - ▶ `operator==`, `!=`
 - ▶ ...

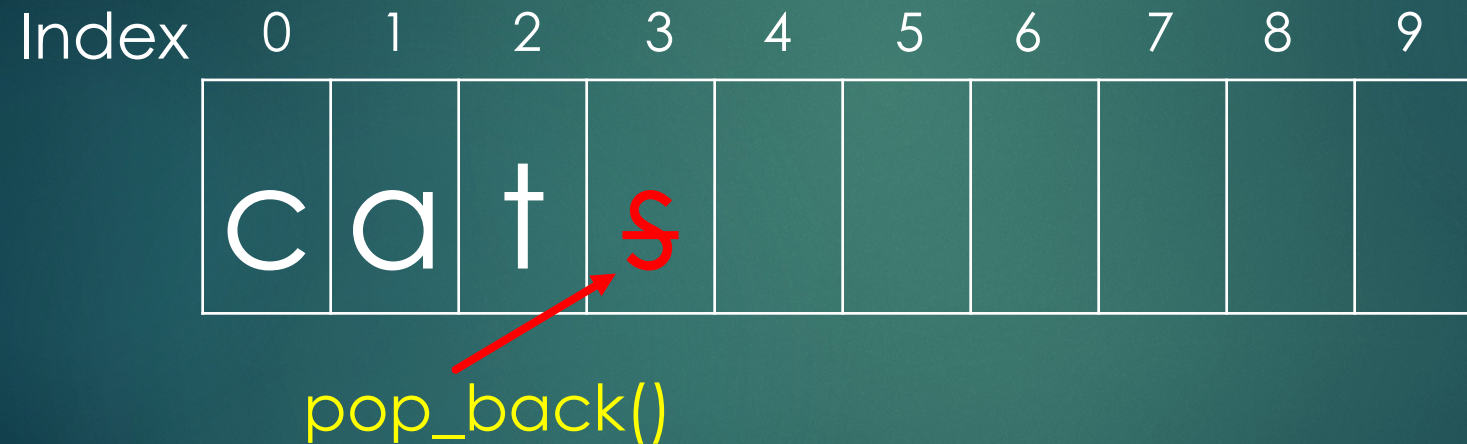
getSize() = 4

- Returns the number of elements in the vector.



getCapacity() = 10

- Returns the capacity of the storage space currently allocated for the vector



getCapacity() = 10

- Returns the capacity of the storage space currently allocated for the vector

Index	0	1	2	3	4	5	6	7	8	9
	c	a	t	m						

push_back('m')

size_t

- ▶ `std::size_t` is the unsigned integer type of the result of the `sizeof` operator as well as the `sizeof...` operator

```
#include <cstddef>
#include <iostream>

int main()
{
    const std::size_t N = 10;
    int* a = new int[N];

    for (std::size_t n = 0; n < N; ++n)
        a[n] = n;
    for (std::size_t n = N; n-- > 0;) // Reverse cycles are tricky for unsigned types.
        std::cout << a[n] << " ";

    delete[] a;
}
```


`void resize(size_t n, char c = '\0')`

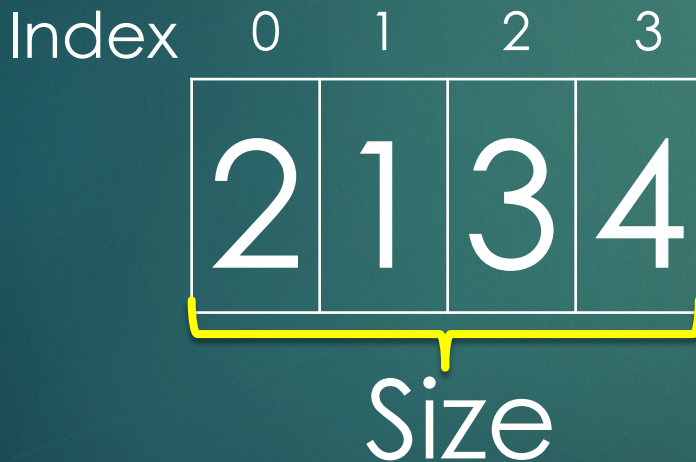
- ▶ Resizes the container so that it contains `n` elements.
 - ▶ If `n` is smaller than the current container `size`, the content is reduced to its first `n` elements, removing those beyond.
 - ▶ If `n` is greater than the current container `size`, the content is expanded by inserting at the end as many elements as needed to reach a size of `n`.
- ▶ The new elements are initialized as `'\0'`
- ▶ If `n` is also greater than the current container `capacity`, an automatic reallocation of the allocated storage space takes place.

`void reserve(size_t n = 0)`

- ▶ Requests that the capacity to be enough to contain `n` characters.
- ▶ If `n` is greater than the current string capacity, the function causes the container to reallocate its storage increasing its capacity to `n`.
- ▶ Otherwise, the function call does NOT cause a reallocation. (the string capacity is not affected)

void shrink_to_fit()

- Requests the container to reduce its "capacity" to fit its "size".



Capacity = 4 = size

What is the *MyString*?

- ▶ Similar to `std::string`, it based on `Cvector` and provide several string operations.
 - ▶ Constructors
 - ▶ `append()`, `substr()`, `insert()`, `erase()`
 - ▶ `find()`, `find_first_of()`
 - ▶ `operator+`
 - ▶ ...

Size & Capacity?

- ▶ A example of input char*
“Happ”:

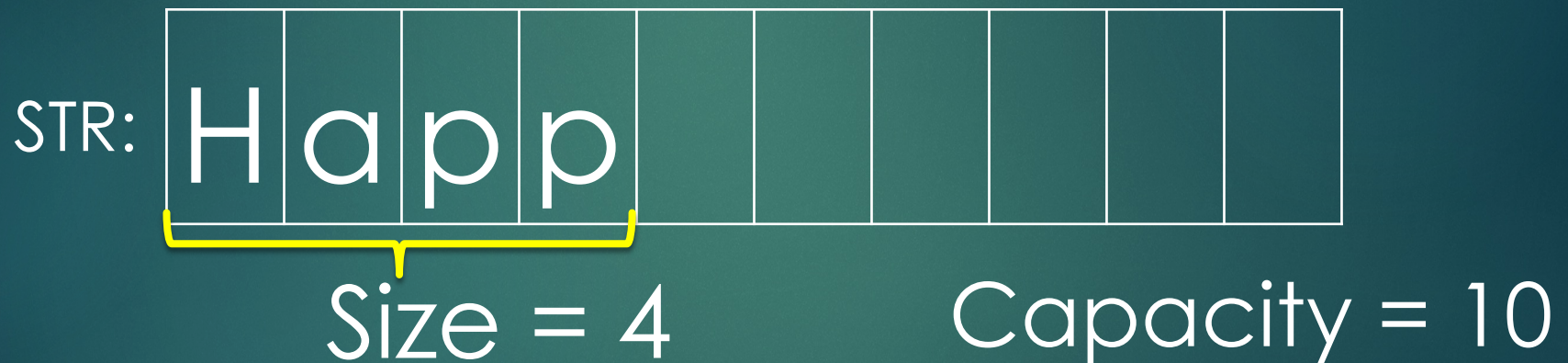
STR:

H	a	p	p	\0
---	---	---	---	----

Null-terminated string

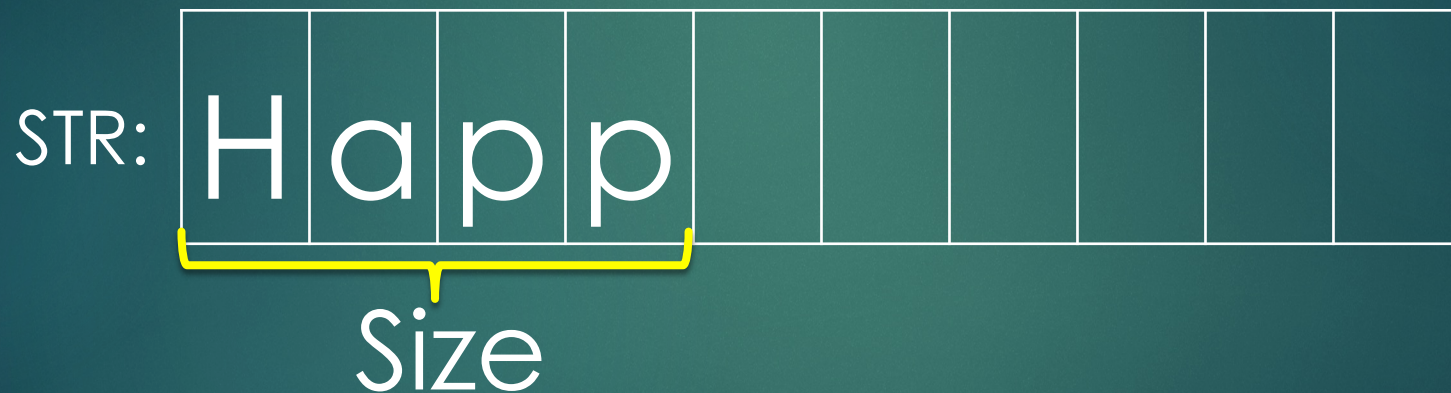
Size & Capacity?

- ▶ A example of our string
“Happ”:



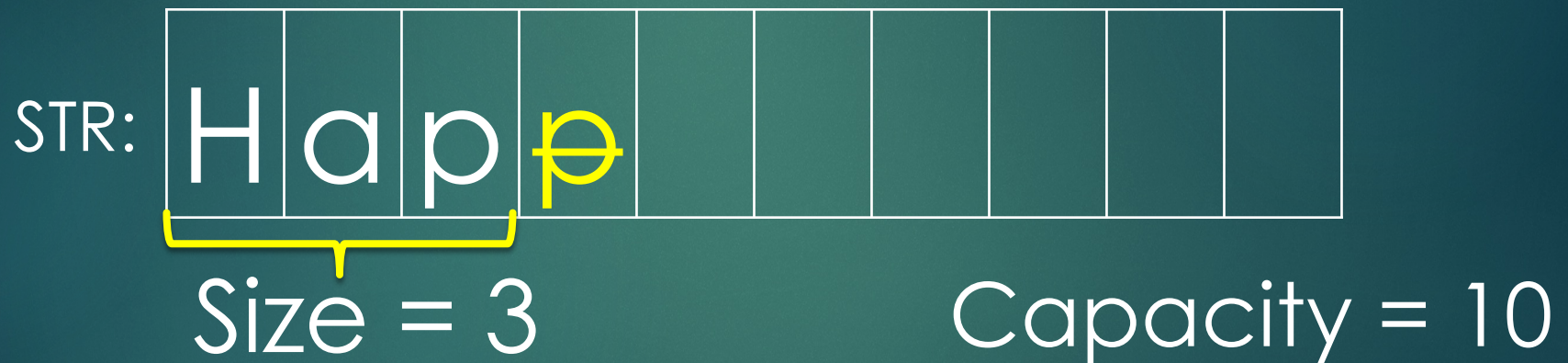
Size & Capacity?

- ▶ A example of our string
“Happ”:



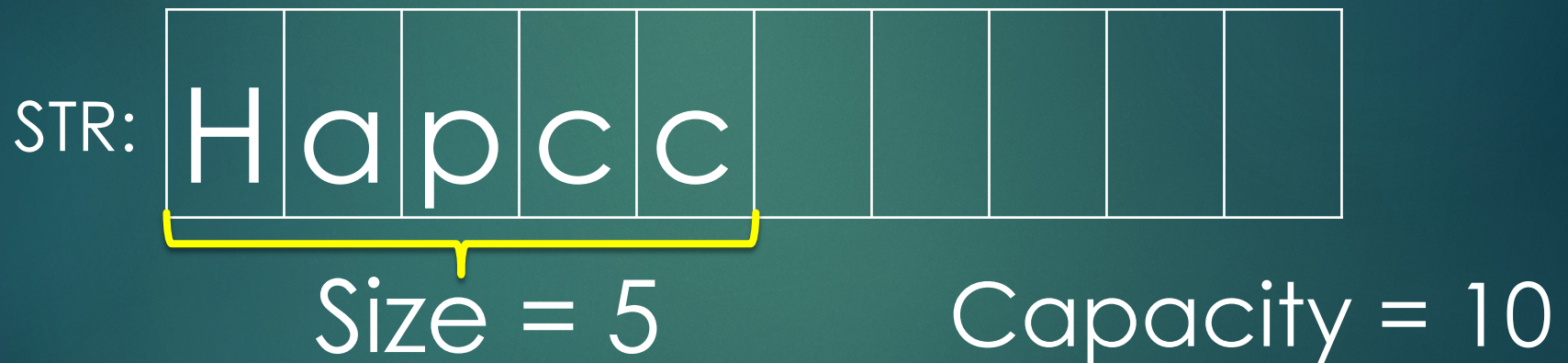
Size & Capacity?

► STR.resize(3)



Size & Capacity?

► `STR.resize(5, 'c')`



Size & Capacity?

► `STR.shrink_to_fit()`

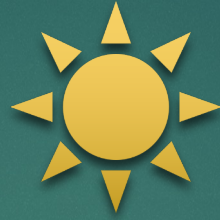


Size = 5

Capacity = 5

Size & Capacity?

► `STR.reserve(7)`

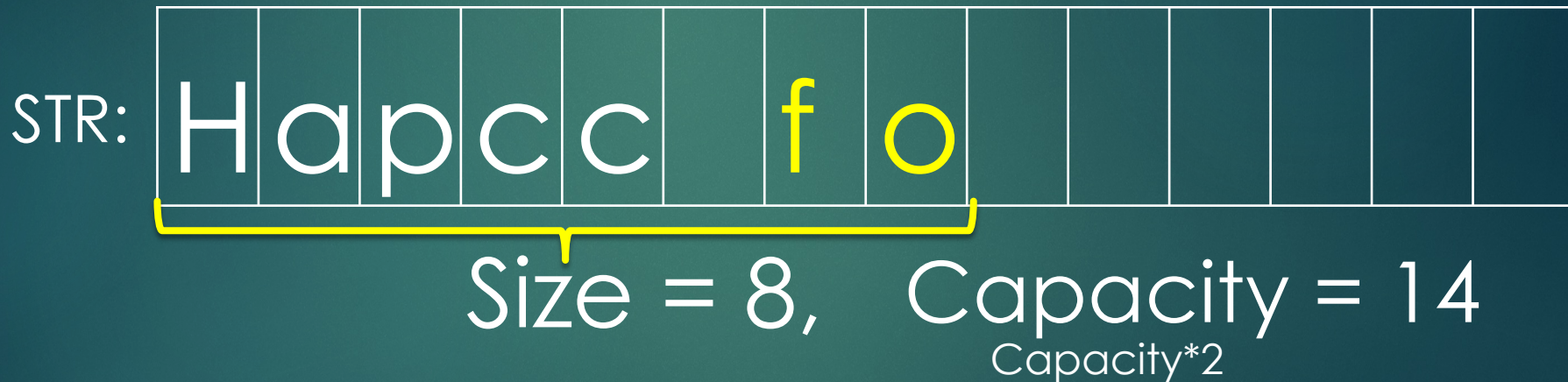


Size = 5

Capacity = 7

Append a string `str2 = "fo"`

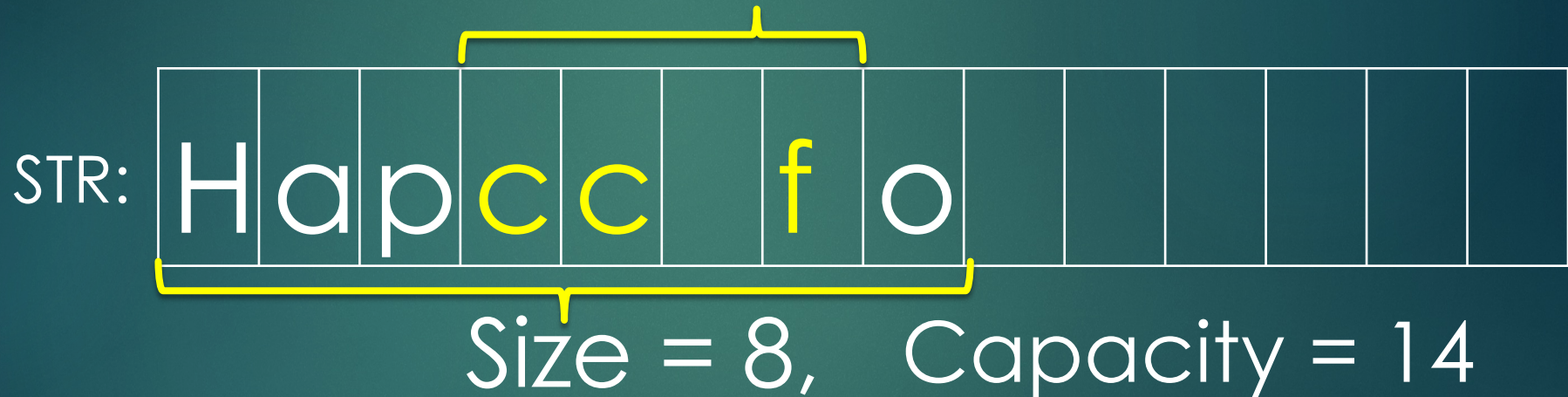
► `STR.append(str2);`



補充說明，因為新追加的字串使得容量(Capacity)不足，
所以Capacity擴增為原來的兩倍

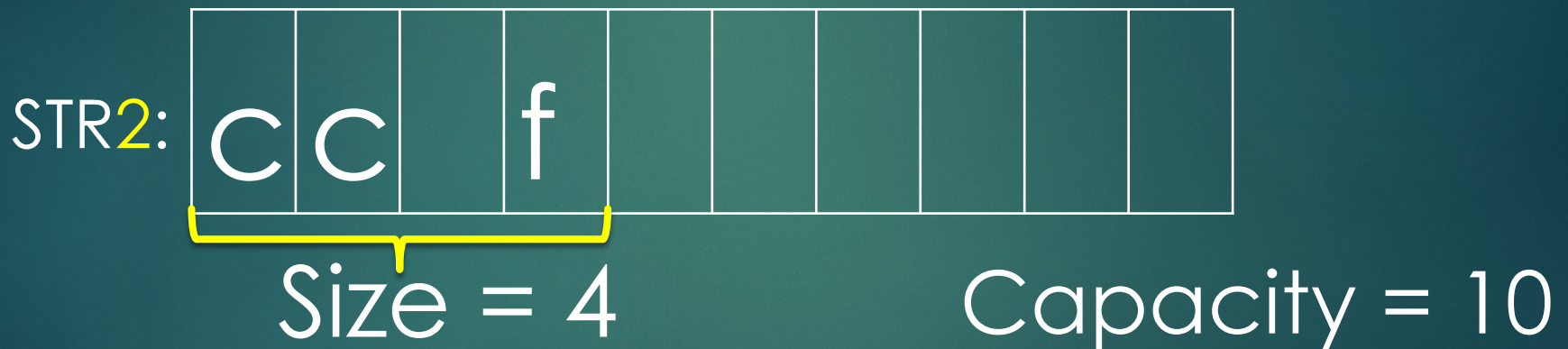
Substr (size_t pos = 0, size_t len = nmax)

► STR2 = STR.substr(3,4);
len = 4



Substr (size_t pos = 0, size_t len = nmax)

► STR2 = STR.substr(3,4);



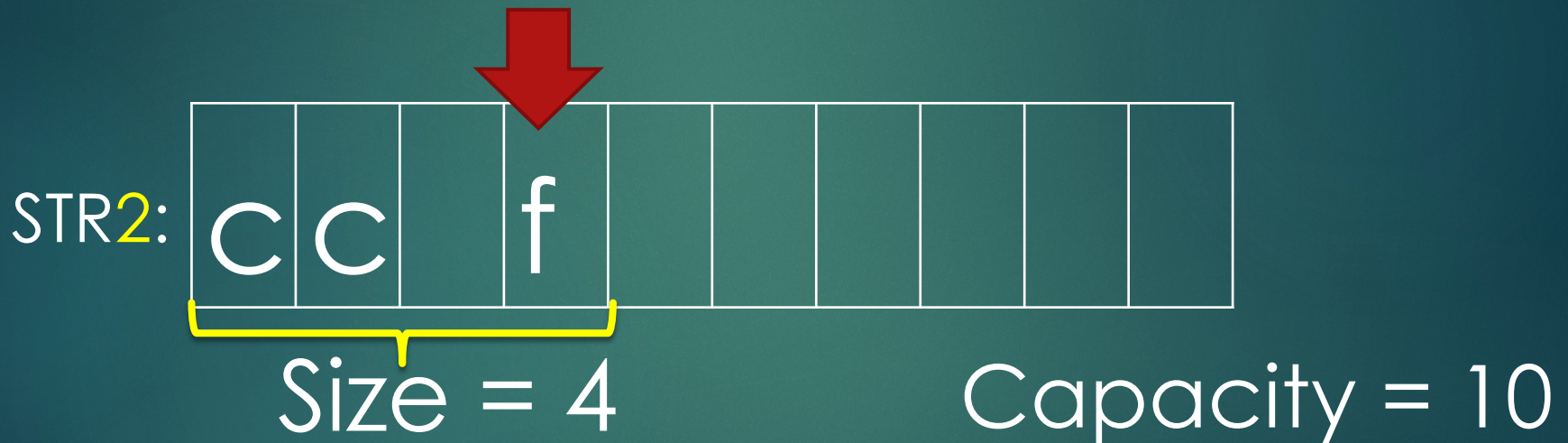
補充說明，因為substr產生新的字串STR2，
因此容量恢復為新產生字串的預設容量（10）

Insert (size_t pos, const String &str,
size_t subpos, size_t sublen),
insert str3 = "abcd"

STEP

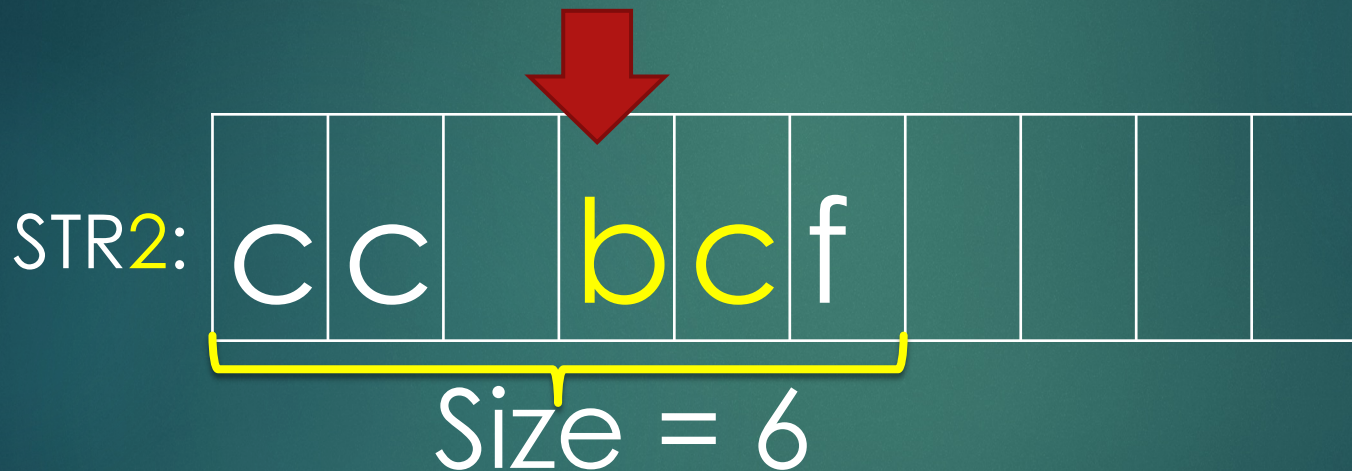
1

► STR2.Insert (3, str3, 1, 2);



Insert (size_t pos, const String &str,
size_t subpos, size_t sublen), insert
str3 = "a**b**c**d**"

► STR2.Insert (3, str3, 1, 2);



find() and find_first_of()

- ▶ STR= "to be or not to be, that is **cool** question"
 0 10 28
- ▶ find "cool" in the STR: **28**
- ▶ find "cXXl" in the STR: **999** //not found
- ▶ find_first_of "cXXL" in the STR: **28**
- ▶ 在STR中找第一個符合 'c', 'X', 'L' 的字元

This time...

- ▶ We will provide two header files and a driver program.
- ▶ Please complete the implementation of these two classes.

Submit your codes to TA

- ▶ Please complete the implementation of the class **CVector** and **MyString** to fulfill all challenge in the driver program (109hw3_main.cpp).
- ▶ Make sure the code could be compiled with our driver program.
 - ▶ If we can't compile your code, you get **0** point too.

Submit your codes to Portal



- ▶ Please use s1234567_CVector.h & .cpp, s1234567_MyString.h & .cpp as your file names.
 - ▶ Replace s1234567 by your own student ID.
 - ▶ And upload **ONLY** these codes.
 - ▶ Please ZIP them with your student ID, s1234567_109hw3.zip
 - ▶ If you try to upload another files (for example *.sln or others), you get 0 point.
- ▶ Then, submit your zip file to Portal.