

CSCI 1380

APRIL 10, 2017

ADMINISTRATIVA

ARRAYS (CONT.)

- Review:
 - `char arr[4];`
 - `char arr[4] = {'a', 'b', 'c', '\0'};`
 - `char arr[] = "abc";`

- array indexing: `arr[i];`
- example (update): `arr[1] = 'a';`
- example (retrieval): `cout << arr[1];`

Example: code that prints out the contents of a character array.

```
char arr[] = "hello!";  
  
for (int i=0; arr[i] != '\0'; ++i)  
{  
    cout << arr[i];  
}
```

Exercise: implement the following function, such that it determines how many characters are in the string being passed to it.

```
int getLength(string s)
```

Exercise: implement the following function, such that it determines whether or not a string is rad...

```
bool isRad(string s)
```

A string is rad if:

- (a) its first character is a letter,
- (b) the last character is an exclamation point, and
- (c) it contains at least one character that is a 'z' or 'Z'.

```
bool isRad(string s)
{
    char c = s[0];
    if ( !( (c >= 'A' && c <= 'Z') || ( c >= 'a' && c <= 'z') ) )
    {
        return false;
    }

    bool hasZ = false;
    int i = 0;
    while (s[i] != '\0')
    {
        if (s[i] == 'Z' || s[i] == 'z')
        {
            hasZ = true;
        }
        ++i;
    }
    if (!hasZ)
    {
        return false;
    }

    c = s[i-1];
    if ( !(c == '!') )
    {
        return false;
    }

    return true;
}
```


Exercise: declare an array to store the letters of the alphabet.

Exercise: write code that prints the contents of the array to the console, with each element printed on its own line.

Exercise: copy the array – in reverse order – into a new array.

- char arrays:
 - `char arr[4];`
 - `char arr[4] = {'a', 'b', 'c', '\0'};`
 - `char arr[] = "abc";`

- general declaration: `(data type) (var name)[x];`
 - `int arr[4];`

- declaration w/population:
 - `int arr[4] = {1, 2, 3, 4};`

- array indexing: `arr[i];`
- example (update): `arr[i] = a;`
- example (retrieval): `cout << arr[i];`

Exercise: declare an array to store 10 numbers.
Populate the array with user input.

Exercise: implement the following function, such that it finds the largest number being stored in the array and returns its *value*.

```
double getMax(double a[], int size)
```

Exercise: implement the following function, such that it finds the largest number being stored in the array and returns its *index*.

```
int findMax(double a[], int size)
```

Exercise: implement the following function, such that it sorts the elements in the array in increasing order.

```
void sort(double a[], int size)
```