



Ajuntament  
de Barcelona

# Basic algorithmics

It will help you on interviews!

June 2019



# Credits

[500 data Structures and algorithms practice problems](#)

[Top 10 algorithms interview problems](#)

# Proyecto soluciones

[Descargar RAR](#)



# Methods that can help

```
toCharArray() //get char array of a String
charAt(int x) //get a char at the specific index
length() //string length
length //array size
substring(int beginIndex)
substring(int beginIndex, int endIndex)
Integer.valueOf()//string to integer
String.valueOf()//integer to string
Arrays.sort() //sort an array
Arrays.toString(char[] a) //convert to string
Arrays.copyOf(T[] original, int newLength)
System.arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
```



# Array rotation problem

## Problema:

Rotar un array simple (String, int, char...) sin utilizar un array auxiliar ni librerías externas.

## Puntos a tener en cuenta:

Puedes usar variables auxiliares  
NO puedes usar arrays auxiliares  
Se puede hacer todo en el main



# Array rotation Solution

```
int[] array={1,2,3,4,5,6,7,8,9,10};  
  
System.out.println(Arrays.toString(array));  
  
for (int i = 0; i < array.length/2; i++)  
{  
    int aux = array[i];  
    array[i] = array[array.length-aux];  
    array[array.length-aux]=aux;  
}  
  
System.out.println(Arrays.toString(array));
```



# Prime number problem

## Problema:

Dado un número no negativo, contar y guardar todos los números primos anteriores.

## Puntos a tener en cuenta:

Podéis usar Listas

Hacedlo en un método que retorne los números primos.



# Prime Number solution

```
public static ArrayList<Integer> countPrimes(int n) {
    n = n - 1;
    ArrayList<Integer> primes = new ArrayList<Integer>();
    if (n <= 1)
        return primes;
    if (n == 2) {
        primes.add(2);
        return primes;
    }

    if (n == 3) {
        primes.add(3);
        return primes;
    }
    primes.add(2);
    primes.add(3);
    for (int i = 4; i <= n; i++) {
        boolean isPrime = true;
        for (int p : primes) {
            int m = i % p;
            if (m == 0) {
                isPrime = false;
                break;
            }
        }

        if (isPrime) {
            primes.add(i);
        }
    }

    return primes;
}
```



# Valid palindrome problem

## Problema:

Dado un String, determina si es un palíndromo considerando sólo caracteres alfanuméricos e ignorando mayúsculas o minúsculas.

## Puntos a tener en cuenta:

No uses librerías.

Podéis utilizar los objetos de los tipos primitivos (Character, por ejemplo)  
Hacedlo en un método que retorna un boolean.





# Valid palindrome solution (part 1)

```
public static boolean isPalindrome(String s) {  
    if (s == null || s.isEmpty()) {  
        return true;  
    }  
  
    int start = 0;  
    int end = s.length() - 1;  
    while (start <= end) {  
        Character first = s.charAt(start);  
        Character last = s.charAt(end);  
  
        if (!Character.isLetterOrDigit(first)) {  
            start++;  
            continue;  
        }  
        if (!Character.isLetterOrDigit(last)) {  
            end--;  
            continue;  
        }  
  
        if (Character.isLetterOrDigit(first) && Character.isLetterOrDigit(last)) {  
            if (Character.toLowerCase(first) != Character.toLowerCase(last)) {  
                return false;  
            }  
            start++;  
            end--;  
        }  
    }  
    return true;  
}
```



# Valid palindrome solution (part 2)

```
public static void main(String[] args)
{
    System.out.println("1: "+isPalindrome("Yo voy"));
    System.out.println("2: "+isPalindrome("Somos o no somos?"));
    System.out.println("3: "+isPalindrome("programad en inglés"));
    System.out.println("4: "+isPalindrome("Somos"));
    System.out.println("5: "+isPalindrome("C# manda"));
}
```



# Lenght of Last Word problem

## Problema:

Dado un string con caracteres mayúsculas, minúsculas y espacios, cuenta la cantidad de letras de la última palabra (determinada por un espacio)

## Puntos a tener en cuenta:

Haz un método que entre UN SOLO STRING (ejemplo: “Yo soy leyenda”)  
No uses librerías externas.



# Lenght of Last Word problem

```
public static void main(String[] args) {
    System.out.println(lengthOfLastWord("yo soy leyenda "));
}

public static int lengthOfLastWord(String s) {
    if(s==null || s.length() == 0)
        return 0;

    int result = 0;
    int len = s.length();

    boolean breakLoop = false;
    for(int i=len-1; i>=0; i--){
        char c = s.charAt(i);
        if((c>='a' && c<='z') || (c>='A' && c<='Z')){
            breakLoop = true;
            result++;
        }else{
            if(breakLoop)
                return result;
        }
    }
    return result;
}
```



# Conclusión

Programad en inglés

