

Actividades de aplicación. Realizar las siguientes 6.11, 6.12, 6.13, 6.16, 6.17, 6.18, 6.19 y 6.20.

**6.11.** Escribe un programa descodificador que muestre un texto codificado con el programa realizado en la Actividad resuelta 6.11.

```
1 package com.mycompany.aa_11;
2
3 import java.util.Scanner;
4 public class Aa_11 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         final char conjuntol[] = {'e', 'i', 'k', 'm', 'p', 'q', 'r', 's', 't', 'u', 'v'};
8         final char conjunto2[] = {'p', 'v', 'i', 'u', 'm', 't', 'e', 'r', 'k', 'q', 's'};
9         char codificado[];
10        char decodificado[];
11        String texto;
12        System.out.print("Introduzca un texto a codificar: ");
13        texto = sc.nextLine();
14        texto = texto.toLowerCase();
15        codificado = new char[texto.length()];
16        decodificado = new char[texto.length()];
17        for (int i = 0; i < texto.length(); i++) {
18            codificado[i] = codifica(conjuntol, conjunto2, c: texto.charAt(index: i));
19        }
20        texto = String.valueOf(data: codificado);
21        System.out.println("El texto CODIFICADO es:\n-" + texto);
22
23        for (int i = 0; i < texto.length(); i++) {
24            decodificado[i] = decodifica(conjuntol, conjunto2, c: texto.charAt(index: i));
25        }
26        texto = String.valueOf(data: decodificado);
27        System.out.println("\n\n---El texto DECODIFICADO es:\n-" + texto);
28    }
29
30    static char decodifica(char conjuntol[], char conjunto2[], char c){
31        return codifica(conjuntol: conjunto2, conjunto2: conjuntol, c);
32    }
33
34
35    static char codifica(char conjuntol[], char conjunto2[], char c) {
36        final String conj1 = String.valueOf(data: conjuntol);
37        char codificado;
38        int pos = conj1.indexOf(ch: c);
39        if (pos == -1) {
40            codificado = c;
41        } else {
42            codificado = conjunto2[pos];
43        }
44        return codificado;
45    }
46 }
47
```

```
--- exec:3.1.0:exec (default-cli) @ Aa_11 ---
Introduzca un texto a codificar:El perro salta por el campo
El texto CODIFICADO es:
-pl mpeeo ralka moe pl caumo

---El texto DECODIFICADO es:
-el perro salta por el campo

BUILD SUCCESS
```

6.12. Realiza el juego del ahorcado. Las reglas del juego son:

- El jugador A teclea una palabra, sin que el jugador B la vea.
- Ahora se le muestra tantos guiones como letras tenga la palabra secreta. Por ejemplo, para «hola» será «\_\_\_\_».
- El jugador B intentará acertar, letra a letra, la palabra secreta.
- Cada acierto muestra la letra en su lugar y las letras no acertadas seguirán ocultas como guiones. Siguiendo con el ejemplo anterior, y suponiendo que se ha introducido: la 'o', la 'j' y la 'a', se mostrará: «\_o\_a».
- El jugador B solo tiene 7 intentos.
- La partida terminará al acertar todas las letras que forman la palabra secreta (gana el jugador B) o cuando se agoten todos los intentos (gana el jugador A).

```
1 package com.mycompany.aa6_12;
2
3 import java.util.Scanner;
4 public class Aa6_12 {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int intentos = 7;
9         System.out.println("*****El juego del ahorcado*****");
10        System.out.println("**Introduzca palabra secreta:\n");
11        String secreta = sc.nextLine();
12        char resultado[] = new char[secreta.length()];
13        for(int i=0; i< secreta.length(); i++){ //inicializo el array secreto
14            resultado[i] = '_';
15        }
16
17        while(!secreta.equals(new String.valueOf(resultado)) && intentos > 0){
18            System.out.println("La secreta es: <<" +
19                String.valueOf(resultado) + ">>");
20            System.out.println("Intentos restantes: " + intentos + "\nIntroduzca una letra: ");
21            char letra = sc.nextLine().charAt(0);
22            boolean acierto = false;
23            for(int i=0; i< secreta.length(); i++){
24                if(secreta.charAt(i) == letra){
25                    resultado[i] = letra;
26                    acierto = true;
27                }
28            }
29            if(!acierto){
30                intentos--;
31            }
32        }
33        System.out.println("----Fin del juego, la palabra secreta era: " + secreta);
34        if(intentos == 0){
35            System.out.println("**** GANA EL JUGADOR A ****");
36        }else{
37            System.out.println("**** GANA EL JUGADOR B ****");
38        }
39    }
40 }
```

```
--- exec:3.1.0:exec (default-cli) @ Aa6_12 ---
*****El juego del ahorcado*****
**Introduzca palabra secreta:
```

```
gato
La secreta es: <<_ _ _>>
Intentos restantes: 7
Introduzca una letra:
a
La secreta es: <<_a _>>
Intentos restantes: 7
Introduzca una letra:
e
La secreta es: <<_a_e>>
Intentos restantes: 6
Introduzca una letra:
i
La secreta es: <<_a_ei>>
Intentos restantes: 5
Introduzca una letra:
o
La secreta es: <<_a_eio>>
Intentos restantes: 5
Introduzca una letra:
u
La secreta es: <<_a_eiou>>
Intentos restantes: 4
Introduzca una letra:
s
La secreta es: <<_a_eious>>
Intentos restantes: 3
Introduzca una letra:
f
La secreta es: <<_a_eiousf>>
Intentos restantes: 2
Introduzca una letra:
d
La secreta es: <<_a_eiousfd>>
Intentos restantes: 1
Introduzca una letra:
c
----Fin del juego, la palabra secreta era: gato
*** GANA EL JUGADOR A ***
```

BUILD SUCCESS

```
--- exec:3.1.0:exec (default-cli) @ Aa6_12 ---
*****El juego del ahorcado*****
**Introduzca palabra secreta:
```

```
gato
La secreta es: <<_ _ _>>
Intentos restantes: 7
Introduzca una letra:
a
La secreta es: <<_a _>>
Intentos restantes: 7
Introduzca una letra:
o
La secreta es: <<_a_o>>
Intentos restantes: 7
Introduzca una letra:
d
La secreta es: <<_a_o_d>>
Intentos restantes: 6
Introduzca una letra:
g
La secreta es: <<ga_o_d>>
Intentos restantes: 6
Introduzca una letra:
t
----Fin del juego, la palabra secreta era: gato
*** GANA EL JUGADOR B ***
```

BUILD SUCCESS

Total time: 27.000 s

6.13. El preprocesador del lenguaje C elimina los comentarios (`/* ... */`) del código fuente antes de compilar. Diseña un programa que lea por teclado una sentencia en C, y elimine los comentarios.

Sentencia: `if (a==3) /* igual a tres */ a++; /* incrementamos a */`  
Salida: `if (a==3) a++;`

```
1 package com.mycompany.aa6_13;
2
3 import java.util.Scanner;
4
5 public class Aa6_13 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Introduzca texto en C:");
9         String texto = sc.nextLine();
10        if(texto.contains("/*")){
11            System.out.println("Texto descomentado: \n" + descomentar(texto));
12        }else{
13            System.out.println("El texto: '" + texto + "' no estaba comentado");
14        }
15    }
16
17    static String descomentar(String texto){
18        String resultado = "";
19        String[] divididoInicio = texto.split("\\/*");
20        for(int i=0; i<divididoInicio.length; i++){
21            if(!divididoInicio[i].contains("/*")){
22                resultado += divididoInicio[i];
23            }else{
24                String[] divididoFin = divididoInicio[i].split("*/");
25                resultado = divididoFin.length > 1 ? resultado + divididoFin[1] : resultado;
26            }
27        }
28        return resultado;
29    }
30 }
31
```

Find: resultado



Previous

Next

Select

aA " . \* [ ]

Output - Run (Aa6\_13) x

Compiling 1 source file with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Aa6\_13 ---

Introduzca texto en C:

`if (a==3) /* igual a tres */ a++; /* incrementamos a */`

Texto descomentado:

`if (a==3) a++;`

BUILD SUCCESS

--- exec:3.1.0:exec (default-cli) @ Aa6\_13 ---

Introduzca texto en C:

`/* COMENTARIO INICIAL*/if (a==3) /* COMENTARIO EN MEDIO 1*/ a++; /* COMENTARIO EN MEDIO 2*/ texto sin comentar al final. /*COMENTARIO AL FINAL*/`

Texto descomentado:

`if (a==3) a++; texto sin comentar al final.`

**6.16.** Lee una palabra o frase y muestra el proceso en el que cada letra se sustituye por otro símbolo no alfabético. Por ejemplo el carácter 'a' se podría sustituir por el carácter '@', la 'e' por '€', la 'i' por '¡', etcétera.

```

1  package com.mycompany.aa6_16;
2
3  import java.util.Scanner;
4
5  public class AA6_16 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9          final char conjunto1[] = {'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o', 'u'};
10         final char conjunto2[] = {'@', '€', '¡', '0', '^', '@', '€', '¡', '0', '^'};
11         char codificado[];
12         char decodificado[];
13         String texto;
14         System.out.print(s: "Introduzca un texto a modificar: ");
15         texto = sc.nextLine();
16         codificado = new char[texto.length()];
17         for (int i = 0; i < texto.length(); i++) {
18             codificado[i] = codifica(conjunto1, conjunto2, c: texto.charAt(index: i));
19         }
20         texto = String.valueOf(data: codificado);
21         System.out.println("El texto MODIFICADO es:\n-" + texto);
22     }
23
24     static char codifica(char conjunto1[], char conjunto2[], char c) {
25         final String conj1 = String.valueOf(data: conjunto1);
26         char codificado;
27         int pos = conj1.indexOf(ch: c);
28         if (pos == -1) {
29             codificado = c;
30         } else {
31             System.out.println("El caracter " + c + " de la posicion " + pos +
32                 " será cambiado por el caracter " + conjunto2[pos] + "");
33             codificado = conjunto2[pos];
34         }
35         return codificado;
36     }
37 }
38
39

```

```

--- exec:3.1.0:exec (default-cli) @ aa6_16 ---
Introduzca un texto a modificaMe gusta pescar peces en el Teide y cazar conejos en el mar
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'u' de la posicion 9 será cambiado por el caracter '^'
El caracter 'a' de la posicion 5 será cambiado por el caracter '@'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'a' de la posicion 5 será cambiado por el caracter '@'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'i' de la posicion 7 será cambiado por el caracter '¡'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'a' de la posicion 5 será cambiado por el caracter '@'
El caracter 'a' de la posicion 5 será cambiado por el caracter '@'
El caracter 'o' de la posicion 8 será cambiado por el caracter '0'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'o' de la posicion 8 será cambiado por el caracter '0'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'e' de la posicion 6 será cambiado por el caracter '€'
El caracter 'a' de la posicion 5 será cambiado por el caracter '@'
El texto MODIFICADO es:
-M€ g^st@ p€sc@r p€c€s €n €l T€ld€ y c@z@r c0n€j0s €n €l m@r
-----
BUILD SUCCESS

```

Entiendo que se refiere a algo así, aunque también se podría hacer algo similar sumandole 101 al valor de cada caracter y que modifique todos los caracteres introducidos.

**6.17.** Construir un programa que convierta una palabra en secuencias de  $n$  letras. Por ejemplo, la palabra «destornillador», dividida en secuencias de 4 letras, se mostrará de la siguiente forma:

```
dest
orni
llad
or
```

```

5 public class Aa6_17 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Introduzca palabra a dividir: ");
9         String palabra = sc.nextLine();
10        System.out.println("Cada cuantas letras desea dividirlo: ");
11        int letras = sc.nextInt();
12        dividir(palabra, letras);
13    }
14
15    static void dividir(String palabra, int letras){
16        String[] dividida = new String[(palabra.length() / letras) + 1];
17        int longitud = palabra.length();
18        int indice = 0;
19        for(int i = 0; indice < longitud; i++){
20            if(indice + letras < longitud - 1){
21                dividida[i] = palabra.substring(indice, indice + letras);
22                indice += letras;
23            }else{
24                dividida[i] = palabra.substring(indice, longitud);
25                indice += letras;
26            }
27            System.out.println(dividida[i]);
28        }
29    }
30 }

```

Output X

Debugger Console X    Run (Aa6\_17) X

```

--- exec:3.1.0:exec (default-cli) @ Aa6_17 ---
Introduzca palabra a dividir:
El caballo blanco de Santiago.
Cada cuantas letras desea dividirlo:
3
El
cab
all
o b
lan
co
de
San
tia
go.
-----
BUILD SUCCESS

```

```

--- exec:3.1.0:exec (default-cli) @ Aa6_17 ---
Introduzca palabra a dividir:
El caballo blanco de Santiago.
Cada cuantas letras desea dividirlo:
5
El ca
ballo
blan
co de
Sant
iago.
-----
BUILD SUCCESS

```

```

--- exec:3.1.0:exec (default-cli) @ Aa6_17 ---
Introduzca palabra a dividir:
El caballo blanco de Santiago.
Cada cuantas letras desea dividirlo:
2
El
c
ab
al
lo
b
la
nc
o
de
S
an
ti
ag
o.
-----
BUILD SUCCESS

```

**6.18.** Escribe una aplicación que convierte una frase (que puede estar escrita con cualquier combinación de mayúsculas y minúsculas) en el nombre de una variable que utilice la nomenclatura Camel. Por ejemplo, la frase «Me GUsTa merenDAR gaLLEtas», se convertirá en «meGustaMerendarGalletas».

Supondremos que cada palabra que compone la frase está separada por un único espacio en blanco.

```
1 package com.mycompany.aa6_18;
2
3 import java.util.Scanner;
4
5 public class Aa6_18 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Introduzca frase para crear la variable camelCase: ");
9         String frase = sc.nextLine();
10        String resultado = "";
11        char[] caracteres = frase.toCharArray();
12        for(int i=0; i<caracteres.length; i++){
13            if(caracteres[i] == ' '){
14                resultado += Character.toUpperCase(caracteres[i + 1]);
15                i++;
16            }else{
17                resultado += Character.toLowerCase(caracteres[i]);
18            }
19        }
20        System.out.println("resultado:");
21    }
22 }
23
```

Output - Run (Aa6\_18) x

```
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBeansProjects\Aa6_18
--- compiler:3.11.0:compile (default-compile) @ Aa6_18 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes
--- exec:3.1.0:exec (default-cli) @ Aa6_18 ---
Introduzca frase para crear la variable camelCase:
Me gUsTa MerendaR CON persONAS.
meGustaMerendarConPersonas.
-----
BUILD SUCCESS
-----
Total time: 3.775 s
```

**6.19.** Implementa un sencillo editor de texto que, una vez que se ha introducido el texto, permita reemplazar todas las ocurrencias de una palabra por otra.

```
1 package com.mycompany.aa6_19;
2
3 import java.util.Scanner;
4
5 public class Aa6_19 {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         System.out.println(x: "Introduzca texto: ");
10        String texto = sc.nextLine();
11        System.out.println(x: "Introduzca palabra a sustituir: ");
12        String sustituida = sc.nextLine();
13        System.out.println(x: "Introduzca palabra sustituta: ");
14        String sustituta = sc.nextLine();
15        texto = texto.replaceAll(regex: sustituida, replacement:sustituta);
16        System.out.println("Resultado:\n" + texto);
17    }
18 }
19
```

Find:  Previous Next Select aA " . \*

Output - Run (Aa6\_19) x

```
--- exec:3.1.0:exec (default-cli) @ Aa6_19 ---
Introduzca texto:
dia 1, dia 2, dia 3, dia 5, otro dia
Introduzca palabra a sustituir:
dia
Introduzca palabra sustituta:
mes
Resultado:
mes 1, mes 2, mes 3, mes 5, otro mes
-----
BUILD SUCCESS
```

**6.20.** Implementa un programa que lea una frase y muestre todas sus palabras ordenadas de forma alfabética. Suponemos que cada palabra de la frase se separa de otra por un único espacio.

```
1 package com.mycompany.aa6_20;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class Aa6_20 {
7
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        System.out.println(x: "Introduzca texto: ");
11        String texto = sc.nextLine();
12        String[] palabras = texto.toLowerCase().split(regex: " ");
13        Arrays.sort(a: palabras);
14        System.out.println(x: "Resultado:");
15        for(String palabra : palabras){
16            System.out.println(x: palabra);
17        }
18    }
19 }
20
```

Find: Previous Next Select

Output x

Debugger Console x Run (Aa6\_20) x

```
--- exec:3.1.0:exec (default-cli) @ Aa6_20 ---
Introduzca texto:
El otro dia fui a pescar
Resultado:
a
dia
el
fui
otro
pescar
-----
BUILD SUCCESS
```