

## Ejercicios de Programación

1. Dada la siguiente frase *"Esta es una frase completa de principio a fin"*, obtener dos Strings, uno que contenga sólo las palabras impares y otro las pares. Al final mostrar el String original, el de las palabras impares y el de las pares. La primera palabra es la que ocupa la posición 1. Solución requerida:

```
Esta es una frase completa de principio a fin
Esta una completa principio fin
es frase de a
```

### Solución:

```
String frase2 = "Esta es una frase completa de principio a
fin";
```

```
String frase2impares="";
```

```
String frase2pares="";
```

```
String aux;
```

```
char character;
```

```
boolean impar = true;
```

```
int indice1=0;
```

```
int indice2=0;
```

```
for (int j=0; j<frase2.length(); j++) {
```

```
    character = frase2.charAt(j);
```

```
    if (character == ' ') {
```

```
        indice2= j;
```

```
        aux = frase2.substring(indice1, indice2);
```

```
        indice1=indice2;
```

```
        if (impar==true) {
```

```
            frase2impares= frase2impares + aux;
```

```
            impar=false;
```

```
        }else {
            frase2pares= frase2pares + aux;
            impar=true;
        }
    }

    if (j==frase2.length()-1) {
        indice2= j+1;
        aux = frase2.substring(indice1, indice2);
        if (impar==true) {
            frase2impares= frase2impares + aux;

        }else {
            frase2pares= frase2pares + aux;
        }
    }
}

System.out.println(frase2);
System.out.println(frase2impares);
System.out.println(frase2pares);
```

2. Dada la frase del ejercicio 1, obtener un String con la misma frase, pero donde los caracteres de las posiciones pares sean mayúsculas y los caracteres de las posiciones impares minúsculas. Los espacios también cuentan como posición y se toma el carácter 0 como posición par. Solución requerida:

```
Esta es una frase completa de principio a fin
EsTa eS UnA FrAsE CoMpLeTa dE PrInCiPiO A FiN
```

**Solución:**

```
String frase3 = "Esta es una frase completa de principio a fin";
String fraseSoluc = "";
```

```
String aux2="";
char character;

for (int j =0; j<frase3.length(); j++) {
    character = frase3.charAt(j);
    aux2 = aux2 + character;

    if (j%2!=0) {
        aux2 = aux2.toLowerCase();
    }else {
        aux2 = aux2.toUpperCase();
    }
    fraseSoluc = fraseSoluc + aux2;
    aux2 = "";
}
System.out.println(frase3);
System.out.println(fraseSoluc);
```

3. Con la misma frase del ejercicio anterior obtener un String donde las palabras impares queden igual, pero las pares se escriban al revés. La primera palabra es la que ocupa la posición 1. Solución requerida:

```
Esta es una frase completa de principio a fin
Esta se una esarf completa ed principio a fin
```

**Solución:**

```
String frase2 = "Esta es una frase completa de principio a fin";
String fraseSolucion="";
String aux3;
String aux4="";
char character2;
boolean par = false;
int indice11=0;
int indice22=0;
```

```
for (int j=0; j<frase2.length(); j++) {  
    caracter2 = frase2.charAt(j);  
    if (caracter2 == ' ') {  
        indice22= j;  
        if (par==false) {  
            aux3 = frase2.substring(indice11, indice22+1);  
            indice11=indice22;  
            fraseSolucion= fraseSolucion + aux3;  
            par=true;  
        }else {  
            aux3 = frase2.substring(indice11, indice22+1);  
            indice11=indice22;  
            for (int jj= aux3.length()-2; jj>-1; jj--) {  
                aux4= aux4+aux3.charAt(jj);  
            }  
            fraseSolucion= fraseSolucion + aux4;  
            aux4="";  
            par=false;  
        }  
    }  
    if (j==frase2.length()-1) {  
        indice22= j+1;  
        aux3 = frase2.substring(indice11, indice22);  
        if (par==false) {  
            aux3 = frase2.substring(indice11, indice22);  
            indice11=indice22;  
            fraseSolucion= fraseSolucion + aux3;  
        }else {  
            aux3 = frase2.substring(indice11, indice22);  
            indice11=indice22;
```

```
        for (int jj= aux3.length()-1; jj>-1; jj--) {
            aux4= aux4+aux3.charAt(jj);
        }
        fraseSolucion= fraseSolucion + aux4;
    }
}

System.out.println(frase2);
System.out.println(fraseSolucion);
```

4. Imprimir una cuenta atrás de 15 al 0 donde en lugar de 15 imprima *"Comenzamos la cuenta atrás"*, en lugar del 5 *"Llegando al final"* y en lugar del cero imprima *"¡Listo!"*.

**Solución:**

```
String inicio = "Comenzamos la cuenta atrás";
String llegando = "Llegando al final";
String fin = "¡Listo!";
for (int i=15; i>=0; i--) {
    switch (i){
        case 15: System.out.println(inicio);
                break;
        case 5: System.out.println(llegando);
                break;
        case 0: System.out.println(fin);
                break;
        default: System.out.println(i);
                break;
    }
}
```

5. En el String *"El dipuTAdo dijo: el muy hijo de puta está imPUtado por la amputación de una PUta oreja"* sustituye todas las apariciones de la palabra malsonante *"puta"* por *"\*\*\*\*"* sin afectar al resto de la frase. Solución requerida:

```
El dipuTAdo dijo: el muy hijo de puta está imputado por la amputación de una PUta oreja

El dipuTAdo dijo: el muy hijo de **** está imputado por la amputación de una **** oreja
```

**Solución:**

```
String fraseMalsonanteRep = "El dipuTAdo dijo: el muy hijo de
puta está imputado por la amputación de una PUta oreja" ;
String fraseFinal = "";
String fraseFinalMinusculas = fraseMalsonanteRep.toLowerCase();
int contador = -1;
int posicion = 0;
int resultado = 0;

do {
    resultado = fraseFinalMinusculas.indexOf(" puta ",
posicion);
    if (resultado > -1) {
        fraseFinal = fraseFinal +
fraseMalsonanteRep.substring(posicion, resultado) + " **** ";
        posicion = resultado +6;
    }else {
        fraseFinal = fraseFinal +
fraseMalsonanteRep.substring(posicion);
    }
} while (resultado > -1);
System.out.println(fraseFinal);
```