

## PROBLEMAS TEMA 5

1.- Escribe un programa en Java que lea un número entero e indique si es par.

2.-Escriba un programa que lea un número entero e indique si el número se puede expresar como el cuadrado de dos números enteros. Por ejemplo, el número 9 se puede expresar como el cuadrado de 3, sin embargo, el número 8 no se puede expresar como el cuadrado de dos números enteros. Para resolver este ejercicio se puede utilizar el método sqrt de la clase Math que permite calcular la raíz cuadrada de un número.

3.- Dado el siguiente fragmento de programa, ¿cuál es el resultado final de la variable z?

```
x = 2;
if (x != 3)
{
    if (x == 1)
        z = 1;
    else
        z = x;
}
else
    z = 0;
```

4.- Escriba un programa en Java que lea un número N mayor que 0 y calcule la siguiente suma  $1+2+3 + \dots + N$ .

5.- Después de ejecutar el siguiente fragmento de programa, ¿cuál será el valor final de la variable x

```
int x = 0;
int n = 16;
while (n % 2 == 0)
{
    x = x + n;
    n = n / 2;
}
```

6.- Escriba un programa que calcule el mínimo, el máximo y la media de una lista de números enteros positivos introducidos por el usuario. La lista finalizará cuando se introduzca un número negativo.

7.- ¿Cuántas veces se ejecuta la sentencia System. out. println en el siguiente fragmento de programa?

```

a = 9;
for (int i = 0; i < 100; i++)
if ( ( a % 4 == 0) || (i % 2 == 0)
Sistem.out.println( a + " " + i);

```

- 1.
2. 50
3. 0
4. 25
5. 100

8.- Dado el siguiente fragmento de programa, ¿cuántas veces se ejecuta la sentencia `System.out.print`?

```

for(int i = 1; i < 10; i++)
for(int k = i; k < 10; k++)
System.out.print( i + " " + k);

```

9.- Escriba un programa en Java que lea dos números: *a* de tipo `double` y *b* de tipo `int`. El programa debe calcular  **$a^b$** .

10.- Escriba un programa que solicite del usuario un número *N* y luego muestre por pantalla la siguiente ejecución:

```

1
1 2
1 2 3
1 2 3 4
.....
1 2 3 4 5 6 .....N

```

11.- En el siguiente fragmento de programa, ¿cuántas veces se ejecuta la sentencia `System.out.println`?

```

for (int i = 1; i <= 20; i = i + 2)
{
    a = 0;
    do
    {
        System.out.println(i + " " + a);
        a = a + 1;
    }
    while (a < 10);
}

```

1. 90
2. 100
3. 81
4. 110

12.- Escriba en Java un programa que lea de forma repetida un número  $N$ . Para cada número leído el programa calculará la suma  $1 + 2 + 3 + \dots + N$ . Una vez mostrado el resultado, el programa preguntará al usuario si desea continuar; si se introduce s el programa continuará la ejecución, en caso contrario finalizará.

13.- Escriba un programa que lea un mes en número (1 para enero, 2 para febrero, etc.) y un año e indique el número de días de ese mes. Recuerde que un año es bisiesto si es divisible por cuatro, excepto cuando es divisible por 100, a no ser que sea divisible por 400. Así, 1900 no fue bisiesto, pero el año 2000 sí lo fue.

14.- Escriba un programa que calcule el número  $n$  con cuatro cifras decimales. Para su cálculo se puede utilizar la siguiente expresión:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)}$$

15.- Un número perfecto es un entero positivo igual a la suma de sus divisores propios. Un divisor propio es un entero positivo distinto que el número en sí mismo, que divide al número de forma exacta (es decir, sin resto). Por ejemplo, 6 es un número perfecto, porque la suma de sus divisores propios 1, 2 y 3 es igual a 6. 8 no es un número perfecto, porque la suma de sus divisores propios,  $1 + 2 + 4$  es distinto de 8. Escriba un programa que acepte un entero positivo y determine si es un número perfecto. Igualmente, muestre todos los divisores propios del número.

17.- Escriba un programa que lea un número entero y lo descomponga en factores primos. Ejemplo:

$$18 = 2 * 3 * 3$$

$$11 = 11$$

$$35 = 5 * 7$$

$$40 = 2 * 2 * 2 * 5$$

18.- Escriba un juego de adivinanza. El programa pedirá al usuario dos números (el número inferior y el número superior), por ejemplo 1 y 100, y un número de intentos, por ejemplo, 4. El programa obtendrá a continuación un número aleatorio entre 1 y 100, y el usuario deberá adivinarlo utilizando como mucho 4 intentos. Cada vez que el usuario introduce un número, el programa le dice si es mayor o menor. Al final, el programa indica si se ha ganado o no.

Para obtener un número aleatorio se puede utilizar la clase `Random` que permite generar números pseudoaleatorios. Esta clase dispone del método `nextInt` que devuelve un número entero uniformemente distribuido, incluyendo valores positivos y negativos.