Olimpiadas Regionales de Informática
Ciudad Autónoma de Melilla

1. EQUPOS KAHOOT EN EL AULA

Antes de hacer un Kahoot en el aula, en modo equipo, hay que agrupar a los alumnos dependiendo del lugar en el que está sentado el alumno, por bancada, para tener proximidad con el resto del equipo. Nuestro programa asignará alumnos a cada equipo, sin sobrepasar el número de 8 miembros, de forma ordenada desde la primera bancada hasta la última, máximo 10 bancadas, siendo 4 el número máximo de alumnos en cada bancada.

El programa solicitará el número de bancadas del aula, y a continuación el número de alumnos sentados en cada bancada. Los equipos se crearán con bancadas seguidas, y la agrupación de alumnos en equipos será la mas ajustada a 8 alumnos, por orden de bancada. La salida del programa serán los equipos obtenidos, cada uno en forma de lista, con el número de alumnos de cada bancada de forma ordenada.

Input Format

La entrada será numérica (entero positivo), primero para el número de bancadas del aula y a continuación los alumnos sentados en cada bancada, desde la primera a la última:

Por ejemplo, 5 bancadas, la bancada 1 contiene 4 alumnos, la bancada 2 contiene 2 alumnos, la bancada 3 contiene 3 alumnos, la bancada 4 contiene 3 alumnos, la bancada 5 contiene 4 alumnos):

5
4
2
3
3
4

Constraints

1 ≤ bancada ≤ 10
0 ≤ alumnosPORbancada ≤ 4

Output Format

La salida será en forma de lista, indicando cada lista, el número de alumnos de cada bancada hasta completar el equipo.

Por ejemplo, 3 equipos con, 4 y 2, 3 y 3, 4 (no sobrepasan 8 alumnos en cada equipo) :
[4, 2]
[3, 3]
[4]

Sample Input 0

```
8
1
4
4
0
2
2
4
4
```

Sample Output 0

```
[1, 4]
[4, 0, 2, 2]
[4, 4]
```

2. CALCULADORA DE IVA

Necesitamos una calculadora de IVA, que pasándole dos parámetros, precio base y tipo de impuesto, calcule el precio del artículo con el impuesto incluido. La calculadora comprobará las restricciones de entrada de datos y devolverá el precio con el IVA añadido. Escribe solamente el código de la función.

Input Format
Para ello el precio base será un número positivo con 2 decimales, y el tipo de impuesto o IVA será un número natural positivo que indica el porcentaje del impuesto (habitualmente en España para la mayoría de artículos, el 21%).

Constraints
preciobase > 0
1 < iva < 99
Output Format
El precio devuelto será un número positivo con 2 decimales.

Sample Input 0

```
100
21
```

Sample Output 0

```
121.0
```

3. ARBOL DE NAVIDAD

Escribe un programa que tome un número entero y produzca la siguiente salida:

Input Format

Un número entero que representa el número de capas del árbol (sin incluir el tronco)

Constraints

$2 \leq n \leq 20$

Output Format

Imprime el patrón del árbol de Navidad según la descripción dada.

Sample Input 0

```
4
```

Sample Output 0

```
   *
  ***
 *****
*******
   |
```

4. SERIES DE FIBONACCI

La serie de Fibonacci modela el crecimiento de una población de conejos. Se sabe que una pareja de conejos puede tener dos crías al mes, a partir del tercer mes de nacidos, cuando los conejos alcanzan su edad madura. La forma en que aumenta la población de conejos mes a mes se puede observar en la siguiente tabla:

```
Mes  | Población de conejos
-------------------------
1    | 1
2    | 1
3    | 2
4    | 3
5    | 5
6    | 8
7    | 13
8    | 21
```

Se pide determinar cuántas parejas de conejos se obtienen dado el mes que se desea determinar. La serie de Fibonacci se obtiene sumando los dos términos anteriores, siendo 0 y 1 los dos primeros términos. La serie continúa con 0, 1, 1, 2, 3, 5, 8, 13, 21, y así sucesivamente.

Input Format

El programa solicitará al usuario ingresar el mes deseado como un número entero.

Constraints

El mes deseado será un número entero mayor o igual a 1. El ejercicio, asume que no hay muertes de conejos y que inicialmente se cuenta con una pareja de conejos recién nacida.

Output Format

El programa imprimirá en la pantalla la cantidad de parejas de conejos en el mes deseado.

Sample Input 0

```
2
```

Sample Output 0

```
1
```

5. TRIANGULO DE PASCAL

Escribe un programa que tome un número entero positivo y produzca las primeras filas del Triángulo de Pascal Modificado.

Input Format

Un número entero positivo

Constraints

1 ≤ n ≤ 10

Output Format

Imprime las primeras filas del Triángulo de Pascal Modificado, donde cada número en una fila es el resultado de la suma de sus dos números vecinos inmediatos, y los extremos de cada fila son siempre 1. La salida tiene que ser un array bidimensional. Separa los números con un espacio.
Ejemplo:

Para n = 5 , la salida esperada sería:

```
1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1
```

Sample Input 0

```
5
```

Sample Output 0

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

6. DIVISIBILIDAD

El programa verifica si el resto de la división A/B es igual a cero para determinar si A es divisible por B.

Input Format

El programa solicitará al usuario ingresar dos números enteros, A y B.

Constraints

Los números ingresados por el usuario serán enteros.

Output Format

El programa imprimirá en la pantalla si A es divisible por B o no con un "Si" o "No"

Sample Input 0

```
4
2
```

Sample Output 0

```
Si
```

7. SUMA DE NUMEROS PARES

El programa itera sobre los números pares desde 2 hasta "n" (inclusive) con un paso de 2, acumulando la suma de estos números.

Input Format

El programa solicitará al usuario ingresar un número entero positivo "n".

Constraints

El número ingresado por el usuario será un entero positivo.

Output Format

El programa imprimirá la suma de los números pares desde 1 hasta "n".

Sample Input 0

```
4
```

Sample Output 0

```
6
```

8. PAR O IMPAR

El programa evalúa si el número es divisible por 2 sin dejar residuo. Si es así, se considera par; de lo contrario, se considera impar.

Input Format

El programa solicitará al usuario que ingrese un número entero.

Constraints

El número ingresado por el usuario será un número entero.

Output Format

El programa imprimirá en la pantalla si el número ingresado es "Par" o "Impar".

Sample Input 0

```
4
```

Sample Output 0

```
Par
```

9. MATRIZ SIMÉTRICA

Dada una matriz cuadrada de tamaño n x n, tu tarea es implementar una función que determine si la matriz es simétrica. Una matriz es simétrica si es igual a su matriz traspuesta.

Para ello, implementa la función `es_simetrica(matriz)`, donde:

- `matriz`: Una lista de listas que representa la matriz cuadrada. Cada lista interna contiene n enteros, representando una fila de la matriz.

La función debe devolver `True` si la matriz es simétrica y `False` en caso contrario.
Ejemplo de Matriz y su traspuesta:

Input Format

- La primera línea contiene un entero , el tamaño de la matriz (n x n).
- Las siguientes líneas contienen enteros cada una, representando los elementos de la matriz.

Constraints

- $1 \leq x \leq 10$
- Los elementos de la matriz son enteros en el rango −1000 a 1000.

Output Format

Imprime "YES" si la matriz es simétrica y "NO" en caso contrario.

Sample Input 0

```
2
1 2
2 1
```

Sample Output 0

```
YES
```

Explanation 0

La matriz y su traspuesta coinciden.

Sample Input 1

```
2
1 2
3 4
```

Sample Output 1

```
NO
```

Explanation 1

La matriz y su traspuesta no coinciden.

10. ¡NOS VAMOS DE REBAJAS!

El programa debe solicitar al usuario el valor total de la compra y luego aplicar el descuento correspondiente según la siguiente tabla:

Compra menor a 10€: Descuento del 5% Compra entre 10€ y 20€: Descuento del 20% Compra mayor a 20€: Descuento del 40% El programa debe calcular el valor a pagar después de aplicar el descuento y mostrar el resultado.

Input Format

El programa solicitará al usuario que ingrese el valor total de la compra como un número entero.

Constraints

El valor de la compra (ingresado por el usuario) será un número entero positivo.

Output Format

El programa imprimirá en la pantalla el valor a pagar después de aplicar el descuento, redondeando el resultado a dos decimales si es necesario.

Sample Input 0

```
15.0
```

Sample Output 0

```
12.0
```