

```
package com.example.actividad_m5_02;

import android.os.Bundle;
import android.util.Log;

import androidx.appcompat.app.AppCompatActivity;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "CalorieCounter";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        try {
            // Leer archivo desde la carpeta raw
            InputStream inputStream =
                getResources().openRawResource(R.raw.input);
            String fileContent = readFileContent(inputStream);

            // Calcular las calorías del elfo con mayor cantidad
            int maxCalories = calculateMaxCalories(fileContent);

            // Imprimir resultado en Logcat
            Log.d(TAG, "El elfo con más calorías lleva: " + maxCalories + " calorías.");
        } catch (IOException e) {
            Log.e(TAG, "Error al leer el archivo: " + e.getMessage());
        }
    }

    // Método para leer el contenido del archivo
    private String readFileContent(InputStream inputStream) throws IOException {
        StringBuilder fileContent = new StringBuilder();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream));
        String line;
        while ((line = reader.readLine()) != null) {
            fileContent.append(line).append("\n");
        }
        reader.close();
        return fileContent.toString();
    }
}
```

```
// Método para calcular el máximo de calorías
private int calculateMaxCalories(String inputData) {
    String[] lines = inputData.split("\n");
    List<Integer> elfCalories = new ArrayList<>();
    int currentCalories = 0;

    for (String line : lines) {
        if (line.trim().isEmpty()) {
            elfCalories.add(currentCalories);
            currentCalories = 0;
        } else {
            currentCalories += Integer.parseInt(line.trim());
        }
    }

    // Añadir las calorías del último elfo
    if (currentCalories > 0) {
        elfCalories.add(currentCalories);
    }

    // Encontrar el máximo
    int maxCalories = 0;
    for (int calories : elfCalories) {
        if (calories > maxCalories) {
            maxCalories = calories;
        }
    }

    return maxCalories;
}
}
```

```
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
```