Sorry about that, here's the same information in markdown:

## Forms and Algorithm for Banker Algorithm

### Form 1

Contains all GUI controls needed to take user input. First, it takes the number of resources, and then it takes the number of processes.

### Form 2

It takes the maximum needed resources.

### Form 3

It takes the current allocated resources. Note that available resources are calculated from total resources minus current allocated resources. Also, current allocated processes must be smaller than total resources and smaller or equal to the maximum needed resources.

### Form 4

It takes the total resources. Note that available resources are calculated from total resources minus current allocated resources.

### Form 5

It represents all given data by the user and contains a button (check) to evaluate the banker algorithm.

### Algorithm

1. Initialize a variable called `safe` to `true` for our `while` loop to iterate among all given data and a boolean array called `processfinished` to check in the end if it's safe or unsafe.

2. For each process:

- Check if the process is finished.

- If the process is finished, skip to the next process.

- If the process is not finished, loop through all the resources:

- Calculate the difference between the maximum needed and allocated resources.

- If the difference is greater than the available resources, the system is in unsafe mode, and we display a `messageBox` declaring the specific resource.

- If the difference is smaller than or equal to the available resources, we will increase the available resources and mark in the `processfinished` array to be `true`.

3. Finally, after passing all iterations, we will decide if it's in safe or unsafe mode by checking the `processfinished` array to be all `true`, else it will be in unsafe mode.

This algorithm is commonly known as the banker's algorithm and is used to avoid deadlocks in operating systems.