Homework Problem 24. Reproduce the proof of the halting problem in all the necessary detail.

Solution. We want to prove that no algorithm is possible that, given a program $p$ and data $d$, checks whether $p$ halts on $d$.

We will prove it by contradiction. Let us assume that there exists an algorithm – i.e., a Java program – halt-checker that:

- *given* two strings: a program $p$ and data $d$,
- *returns* true if $p$ halts on $d$, and false otherwise.

Let us now build the following auxiliary program:

```java
public state int aux(String x){
    if (halt-checker(x,x))
        {while(True) x = x;}
    else {return 0;}}
```

Will this program aux halt if, as input, we give it the same string aux?

- If aux *halts* on aux, then, by definition of the halt-checker, the value
  
  \[
  \text{halt-checker(aux,aux)}
  \]

  is true. If you trace the above program aux, you will see that in this case, this program will go into an infinite loop – and thus, it will *not* halt.

- On the other hand, if aux *does not halt* on aux, then, by definition of the halt-checker, halt-checker(aux,aux) is false. If you trace the above program aux, you will see that in this case, this program *will halt* – namely, it will return 0.

In both case, we get a contradiction:

- if we assume that it halts, we conclude that it does not halt, and
- if we assume that it does not halt, then we conclude that it halts.

This contradiction shows that halt-checkers are indeed not possible.