github.com/ljhao288/Go_practice

Task 1

Add an id for each worker so that when printing the result, we can know which worker processed this job.
Based on “jobForWorker_v5”
Hint:
Add an id property in worker struct

```
16  type Worker struct {
17      id          int
18      WorkerPool chan chan string
19      JobChannel chan string
20  }
```

Expect output:

```
PS C:\NonSystemFile\WorkSpace_go\src\jobForWorker_v5_1> go run main.go
8
Setting success
Worker 3: http://fake9.comis up!
Worker 4: http://fake10.comis up!
Worker 9: http://fake14.comis up!
Worker 8: http://fake11.comis up!
Worker 1: http://fake7.comis up!
Worker 5: http://fake12.comis up!
Worker 4: http://fake22.comis up!
Worker 9: http://fake17.comis up!
Worker 1: http://fake27.comis up!
Worker 5: http://fake23.comis up!
Worker 9: http://fake28.comis up!
Worker 4: http://fake35.comis up!
Worker 4: http://fake29.comis up!
Worker 9: http://fake40.comis up!
Worker 5: http://fake31.comis up!
Worker 5: http://fake41.comis up!
Worker 4: http://fake42.comis up!
Worker 9: http://fake37.comis up!
Worker 4: http://fake43.comis up!
Worker 5: http://fake39.comis up!
Worker 9: http://fake44.comis up!
```

Task 2

Let the program exit after 10 seconds
Based on “jobForWorker_v5”

Task 3

Based on “lbExample_v2”
Why this program works?
Try to understand the readAndWritePort function. Does it read and write to channel at the same goroutine and have a contradiction with what we discussed before?