INSTALL & TEST

1, environment:

```
xinu08 218 $ uname -a
Linux xinu08.cs.purdue.edu 3.12.6 #1 SMP Wed Feb 12 16:26:06 EST
2014 x86_64 Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz GenuineIntel
GNU/Linux
```

2, folder content:

```
xinu08 223 $ cd forestFire/
xinu08 224 $ ls
dataStruct.h  main.c  nextStage.c  r
```

3, compile:

```
xinu08 225 $ gcc -o test main.c
xinu08 226 $ gcc -o nextStage nextStage.c
```

please make sure to use “nextStage” while compiling nextStage.c

4, run:

(a) input format

```
xinu08 227 $ ./test
[error]: Not enough arguments. <processes> <iterations> <pb>
<pa> <output freq>
```

(b) testcase: 2 processes, 30 iterations, pb 0.0001, pa 0, freq 10.

```
xinu08 234 $ ./test 2 30 0.0001 0 10
[main]: start to create child processes
 ...
[main]: All processes have finished
 simulation results are under ./r
```

(c) result:

```
xinu08 235 $ cd r/
xinu08 236 $ ls
itr10.txt  itr20.txt  itr30.txt
```

 zheng178@purdue.edu

FEATURE

1, Tolerant for invalid user inputs.

2, pa and pb has accuracy of 0.0001 during calculation. Accuracy could be changed from dataStruct.h, ACCU macro.

3, if <iter> modulo <freq> is not zero, the last iteration is guaranteed to be output to a file.

DESIGN

1, Shared memory:
    main.c allocates, initializes, and deallocates shared memory.
    Each child process attaches shared memory segment to its own address space.

2, Semaphores:
    main.c creates, initializes, and deletes semaphore.
    Semaphore has value 1, which only allows one process to modify shared memory.

3, Pipeline:
    Each child process has 2 stages: (a) modify shared memory, (b) output to file.
    (a) has to be atomic. (b) can be overlapped.
    Thus, I consider multi processes could run faster than single process.
    (I didn't have time to test though.)