Assignment - 2, Part - 1

1. Write a program that lists all primes between 1 and n where n is input from the keyboard (specified by the user). The main process, P creates two child processes, C1 and C2. P filters each integer between 1 and n that is divisible by 2 or 3 or 5 or 7 or 11 and communicates each unfiltered integer through pipes to either C1 or C2. The child processes perform a primality check on each received integer and output the prime numbers on to the screen. Make sure that the loads on C1 and C2 are roughly balanced.

2. Elections have just been held in over 1000 constituencies. A candidate represents one of 20 political parties. Each of the 200 districts has multiple constituencies. The party-wise number of winning candidates in a district is contained in one of 10 input files. An input file contains information on multiple districts. The file format is shown below.

```
121
13      4
9       1
15      2
16      1
4       1

765
12      3
16      2
7       1
8       2
5       2

. . .
```

The interpretation of the above is - in District 121, Party 13 bagged 4 seats, Party 9 bagged 1 seat, and so on. Note that only parties that bag non-zero seats in a district appear in the input.

Write a program to compute the country-wide, party-wise tally of winning candidates. Create 10 threads – a thread is responsible for processing the input from one of the 10 files. After a thread has read the results pertaining to a district, it updates the tally for each party. Note that the tally for each party is shared and updated by all threads concurrently.

As soon as the (cumulative) number of winning candidates exceeds a multiple of 50, the cumulative "scores" of each party are updated on the screen. An example is shown below (the scores of the 20 parties are shown in order, i.e. Party 1, Party 2, etc.)

```
0 1 0 2 3     0 4 11 2 0     0 5 4 1 7     5 2 3 1 0
```