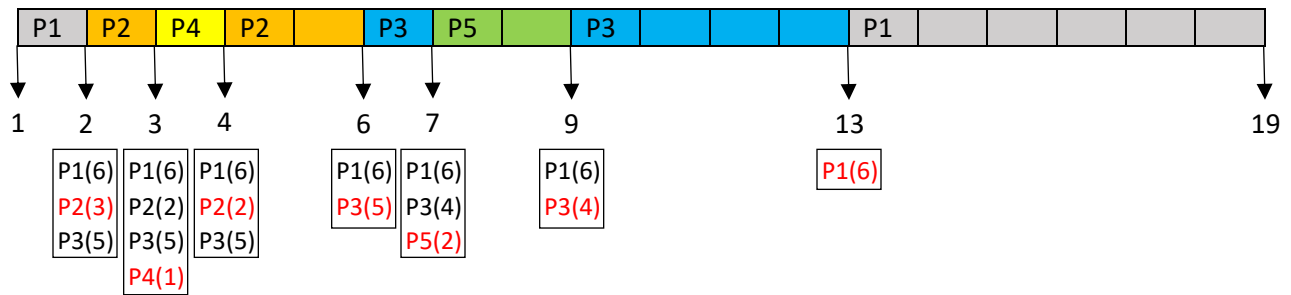


Q1.



$$ART = ((19-1) + (6-2) + (13-6) + (4-3) + (9-7))/5 = 32/5 = 6.4 \text{ sec}$$

$$ART = ((19-1) + (6-2) + (13-2) + (4-3) + (9-7))/5 = 36/5 = 7.2 \text{ sec}$$

Q2.

A.

Got SIGUSR1

Got SIGUSR1

Got SIGUSR1

Got SIGUSR1

Got SIGUSR1

It takes 5 seconds to run.

B.

Almost 0 second. When the parent process receives the first SIGUSR1 from its child process, it will be terminated.

Q3.

1. 12000
2. 12001 (I/O, takes 3 ins.)
3. 100
4. 101
5. 102
6. 103
7. 8000
8. 8001
9. 8002
10. 8003 (I/O, takes 7 ins.)
11. 100
12. 101
13. 102
14. 103
15. 5000
16. 5001
17. 5002
18. 100
19. 101
20. 102
21. 103
22. 5003
23. 5004 (I/O, takes 6 ins.)
24. 100
25. 101
26. 102

27. 103  
28. 8004  
29. 8005  
30. 100  
31. 101  
32. 102  
33. 103  
34. 5005 (I/O, takes 4 ins.)  
35. 100  
36. 101  
37. 102  
38. 103  
39. 8006  
40. 100  
41. 101  
42. 102  
43. 103  
44. 5006  
45. 5007  
46. 5008 (I/O, takes 5 ins.)  
47. 100  
48. 101  
49. 102  
50. 103  
51. 12002  
52. 100  
53. 101  
54. 102  
55. 103  
56. 5009  
57. 5010  
58. 5011  
59. 5012  
60. 100  
61. 101  
62. 102  
63. 103  
64. 12003  
65. 12004  
66. 12005  
67. 12006 (I/O, takes 1 ins.)  
68. 100  
69. 101  
70. 102  
71. 103  
72. 12007  
73. 12008 (I/O, takes 2 ins.)  
74. 100  
75. 101  
76. 102  
77. 103  
78. 12009  
79. 12010

Q4.

Yes, the program can possibly produce the output "2 children are done".

The parent process will receive 5 SIGCHLD almost at the same time, but the arrival time may be **slightly different** for each child process. It depends on process scheduling of the system.

In my OS, the results can be "2 children are done", "3 children are done", "4 children are done", "5 children are done".

Q5.

```
int num_primes = 0; // holds # of primes
void count_primes (int start, int end){
    // check each number in range [start, end] for primality
    for (int num = start; num <= end; ++num) {
        int i;
        for (i = 2; (i <= num) && (num % i != 0); ++i);
        if (i == num) // if no divisor found, it is a prime
            ++num_primes;
    }
}

void signal_handler (int signo){
    cout << "The number of primes found is: \n" << num_primes << endl;
    exit(signo);
}

int main (int ac, char** av){
    signal (SIGINT, signal_handler);
    count_primes (1, 1000000); // count all primes <= 1 million
    cout << "Found " << num_primes << " prime numbers in the range" << endl;
}
```