Assignment-5
Report

180010030,180010032,180020006

Algorithm	No. of cycles	Throughput
Odd-Even	249	0.024
Fibonacci	3157	0.029
Descending	11433	0.031
Prime	1175	0.028
Palindrome	1983	0.028

1) The code for odd-even has no control hazards because the conditional branch was not taken as the value checked was odd. There are a few RAW hazards due to which we see a small number of data hazards.
2) For fibonacci, there is a control hazard in the loop. Every time it goes in the loop, it is a wrong branch, till the count reaches 10. So, we see a few control hazards. Similarly, there are a few data hazards outside the loop and a few inside. Hence, we also see some stalls in the pipeline.
3) The code for descending has a lot of branching, 3 nested loops and data hazards. Hence, we see that the total number of hazards increases by a large amount and so does the cycles executed.
4) For Prime, the loop runs from n=2 to n=5, i.e. 4 times. All these 4 times, we get a wrong branch taken for the jmp statements. Then it takes a wrong branch when it is supposed to enter prime/not prime. Hence, a total of 10. Similarly, we are getting a few data hazards.

5) In this, every time the code misses the jump to loop, there is a wrong branch taken. So we get a control hazard. But we have more data hazards per loop. Hence, we see a larger number of data hazards than control hazards.

^ Here in all cases we can commonly observe that throughput is almost the same as it is basically a proportion of instruction per cycle.