1. Frodo and Sam want to implement an operation $A + B \bmod P$ on FPGA. The size of A,B and P is 255 bits and $A < P, B < P$. This operation is defined as follows:

Algorithm 1 A+B mod P

```
1: T=A+B
2: if T > P then
3:     T=T-P
4: end if
5: Return T
```

Sam has written the following code for this

```
module (A,B,P,out) input
[254:0] A,B,P; output [254:0]
out; wire [255:0] sum_out;
wire [255:0] sub_out; wire
sign;

assign sum_out=A+B; assign {sign,
sub_out}=sub_out-P;
assign out= (sign==1)?sum_out[254:0]:sub_out[254:0]; endmodule
```

However, Gandalf is not happy with this architecture. He has asked Sam to use only a 64 bit adder and a 64 bit subtractor for his architecture. But, he can consume multiple clock cycles to construct his design. Help Sam and Frodo to achieve this objective by writing the required HDL code

- The design should not use more than five cycles to get the correct result.
- For 64-bit adder, please use the fast FPGA adder code uploaded in helloIITK (solution of question no 2 from Assignment 2 of Monday's batch). For subtractor, you can use simple assignment statement.
- You need to verify the design on FPGA using IP.
- You need to compare the design performance by measuring the path delay and compare it with the first code of Sam.