1a)

Why we don’t use float for all our computations: It is not large enough to store the intermediate data of the size that we require, in other words we would lose precision due to truncation.

Why we don’t use double for all our input data: It would cause our input to be much larger and we might run out of space for large inputs, or just not be able to store as large input.

1c)

For E[X] We have (n-1) additions & 1 mult, so total number ops is n.

For the moments, we have (kmax-2)\*(n-1) additions & (kmax-2)\*1 mult to divide by n, and (kmax-1)\*n mults for the values to be added (given that we store the multiplication values and just multiply each value of the input array by the original value).

So the total number of computations is:

n + (kmax-2)\*(n-1) + (kmax-2) + (kmax-1)\*n

= n + kmax\*n - kmax – 2n +2 + kmax-2 + kmax\*n – n

= 2kmax\*n – 2n

= 2n(kmax-1)

1d)

I was able to get 1.30(10^9) FLOPS with n=40000000 and kmax=7.

1e)

Lab machines max throughput: 3470(10^9) FLOPS

My implementation’s throughput: 1.30(10^9) FLOPS

Fraction of peak throughput achieved:

1.30/3470

= 0.0003746397695

2c)



