Assignment #3

Use the “steps” in “program_organization.txt” from the recent lecture in order to solve the following questions. Label each “step” of your program in the same manner as the example5_statistics2 program (eg. 3. INPUT).

1. Use Flowgorithm to write the program for example5_statistics2. Then print out the flow chart and circle / label each “step” in the program (eg. 3. INPUT).

From this result you can see that the program organization steps of the notes are analogous to the Flowgorithm components. The implication is that the these “steps” cover a fairly general class of programs across many different languages in a manner similar to Flowgorithm.

Therefore, this organizational approach is probably a good way to organize the task of programming from a big picture / high level perspective since it’s applicable to most programming languages.

2. Write a program that repeatedly evaluates a n-th order polynomial

p(x) = a0 + a1*x + a2*x^2 + ... + an*x^n

where n <= 10

The program inputs n, ai, and x from the keyboard and then prints out the corresponding value of p to the screen.

The program continues to input new values of n, ai, x and evaluates p until a negative value for n is input, at which point the program stops.

3. Write a program that calculates exp(x) using the Taylor series approximation

exp_approx = 1 + x + x^2 / 2! + x^3 / 3! + ...

where x and error are input from the keyboard and a sufficient number of series terms are used such that abs(exp_approx - exp(x)) < error.

Hint: Use *= in a loop to calculate x^i and i! and print them out to the screen to verify that your loop is correct before calculating the series approximation.