

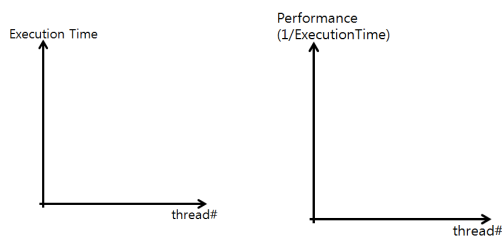
[Problem 2] Parallelize **prob2.c** (see our class webpage project 3 announcement to access **prob2.c**) using OpenMP. Your program should take three command line arguments: scheduling type number (1=static, 2=dynamic, 3=guided), chunk size, and number of threads as program input argument. Your code should print the execution time and the result of PI calculation. Assume the number of steps **num_steps** = 10000000.

command line execution: > **a.out scheduling_type# chunk_size #_of_thread**

execution example> **a.out 2 4 8** <---- this means dynamic scheduling (chunk size = 4) using 8 threads.

(i) submit the OpenMP source code prob2.c

(ii) Write a document (in PDF file format) that reports the parallel performance of your code. Your report should contain (a) following tables and graphs that shows information in the tables, and (b) brief explanation and interpretation on the results (including why such results can be obtained).



execution time (unit:ms)	chunk size	1	2	4	6	8	10	12	14	16
static	1									
dynamic										
guided										
static	5									
dynamic										
guided										
static	10									
dynamic										
guided										
static	100									
dynamic										
guided										

performace (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	1									
dynamic										
guided										
static	5									
dynamic										
guided										
static	10									
dynamic										
guided										
static	100									
dynamic										
guided										

[Problem 3] Create a demo video file (.mp4 format) that shows compilation and execution of your source files (prob1.c, prob2.c). The size of the demo video file should be less than 50MB.