CSE108
LW 11

- **Using mobile phones, flash disks, internet and any other record or communication media is strictly forbidden during lab sessions. Throughout a lab session, all such media must be kept turned off and in a closed environment. Violation of this rule is punished with a grade 0, -100 or worse. Before doing anything else, make sure that your computer is not attached any such media.**
- **Make sure that you have deleted all of your work PARMANENTLY before leaving the first sessions.**

PART 1 (4 Pts)

•Write a complete program for the following @ operation;

$$x \ @ \ y = \begin{cases} x^3 - 2 * x * y, x \geq y \\ y^3 - 2 * x * y, x < y \end{cases}$$

Your program should be able to calculate the value of an expression including @ operation and **print** the value. The program should have the following features:

- Define a macro for the operation
- Define a macro for the print operation.
- Pass the expression as the console arguments to the program..
- Perform verbose print operation when in the DEBUG mode. Define a constant macro (Debug can be zero or one) to see the results step by step with the print operation.

For example;

./test_part1 2 @ 5	
//If debug is 1 , prints • "2 @ 5 equals to 105"	// If debug is 0, prints •"equals to 105"

or ;

./test_part1 11 @ 10 @ 12	
//If debug is 1 , prints • "10 @ 11 equals to 1111 •1111 @ 12 equals to 1371303967 "	// If debug is 0, prints •" equals to 1371303967 "

If the arguments are not in the correct sequence, return a error message and use exit function. For example;

./test_part1 1 @ 2 @ @ 45	
//If debug is 1 , prints • " wrong parameters " ,and exits	• // If debug is 0, Just exits.

PART 2 (3 Pts)

•You will write a library (header file and source file) to handle time operations. Write a test program, as well. You should define a structure type the_time to represent the time including an integer to represent the hour and another one to represent the minute. Your library should also include following functions;

- **void set_time(the_time* time1, int new_hour ,int new_minute)** will set the hour and the minute of the time1 .Note that if the new hour is bigger than 24 or the new minute is bigger than 60 , set as -1.

- **int get_hour(the_time time1)** will return the hour of time1.

- **int get_minute(the_time time1)** will return the minute of the time1.

- **void print_time(the_time time1)** will print the time.

PART 3 (3 Pts)

- **int abs_dif_time(const the_time* time1,const the_time* time2)** will take two times and return the absolute difference in minutes between the times.

- **the_time add_minute(the_time* time1,int add_min)** will take a time and an integer (minutes). It will return the time after adding **add_min** and **time1 .**