

1. 직선은 시작점과 끝점으로 표현할 수 있다. 두 좌표 (x_1, y_1) , (x_2, y_2) 를 전달받아 직선의 기울기 a 와 y -절편값 b 를 구하여 리턴하는 함수 `void LineParameter(int x1, int y1, int x2, int y2, double* a, double* b)`, 두 좌표 (x_1, y_1) , (x_2, y_2) 를 전달받아 직선의 중점의 좌표 (x_3, y_3) 를 구하는 함수 `void Middle(int x1, int y1, int x2, int y2, double* x3, double* y3)`를 작성하시오. 또한 이를 이용하여 입력한 직선의 기울기, y -절편, 중점을 출력하는 프로그램을 작성하시오. 단 $x_1 \neq x_2$ 로 가정하시오.

(Ex. $(1,1) (3,5) \rightarrow \text{Slope:}2 \text{ Y-intercept:}-1$ Middle: $(2,3)$ / $(1,1) (2,3) \rightarrow \text{Slope:}2 \text{ Y-intercept:}-1$ Middle: $(1.5,2)$ / $(1,1) (3,2) \rightarrow \text{Slope:}0.5 \text{ Y-intercept:}0.5$ Middle: $(2,1.5)$)

2. 이차방정식 $ax^2 + bx + c = 0$ 의 실근을 구하여 포인터를 이용하여 반환하고 또한 실근의 개수를 리턴하는 함수 `int GetSolution(int a, int b, int c, double* value1, double* value2)`를 작성하고 이를 이용하여 사용자로부터 세 개의 양의 정수 a, b, c 를 입력 받아 이차방정식 $ax^2 + bx + c = 0$ 의 실근의 수와 실근값을 출력하는 프로그램을 작성하시오. 단 $a \neq 0$ 로 가정하시오.

(Ex. $1 \ 2 \ 3 \rightarrow 0$ / $1 \ 2 \ 1 \rightarrow 1$ -1 / $1 \ -4 \ 3 \rightarrow 2 \ 3 \ 1$ / $2 \ 1 \ -1 \rightarrow 2 \ 0.5 \ -1$)

3. 양의 정수 n 을 전달받아 만약 n 이 소수(prime number)이면 1을 리턴하고, 만일 2가 아닌 짝수인 경우에는 $a + b = n$ 을 만족하는 두 소수 a, b 를 주어진 포인터를 이용하여 반환하고 0을 리턴하는 함수 `int FindPrime(int n, int* a, int* b)`를 작성하고 이를 이용하여 사용자로부터 양의 정수를 입력 받아 그 수가 소수이면 "Prime Number"라는 메시지를 출력하고 짝수일 경우에는 $a + b = n$ 을 만족하는 두 소수 a, b 를 출력하고 그 외의 경우에는 "Try Again"을 출력하는 프로그램을 작성하시오.

(Ex. $255 \rightarrow \text{Try Again}$ / $256 \rightarrow 5 \ 251$ / $257 \rightarrow \text{Prime Number}$)

4. 하나의 정수에 4자리의 십진수를 기억하는 길이가 20인 정수 배열을 이용하여 80자리의 수를 처리하고자 한다. 이 배열에 1000 이하의 정수 n 을 곱하는 함수 `void Multiply(int number[], int n)`을 작성하시오. 단, 이 함수를 호출하면 각 배열에는 0-9999의 값이 저장되어야 합니다. 이 함수를 이용하여 사용자로부터 입력받은 자연수 n 의 팩토리얼 값($=n!$)을 계산하여 출력하는 프로그램을 작성하시오. 최상위의 연속되는 0은 제거하시오.

(Ex. $50 \rightarrow 30414093201713378043612608166064768844377641568960512000000000000$)