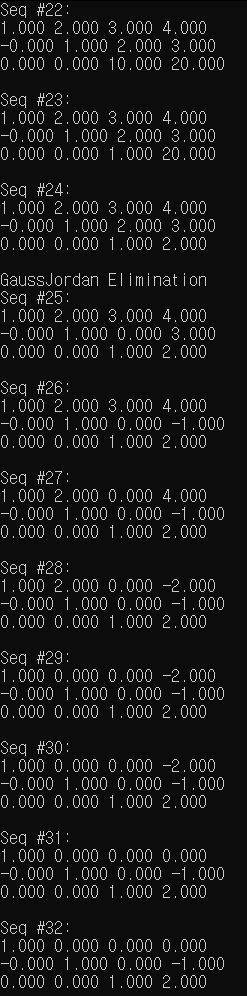
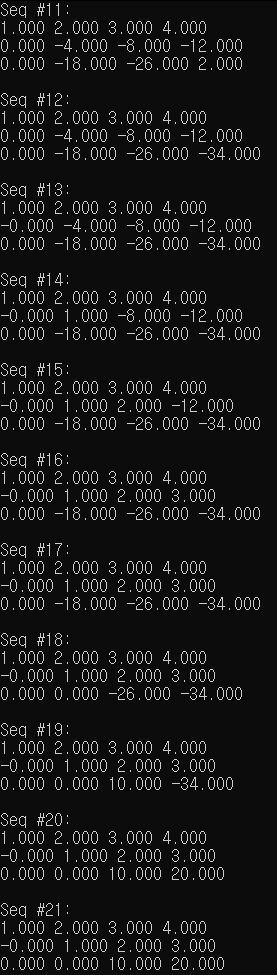
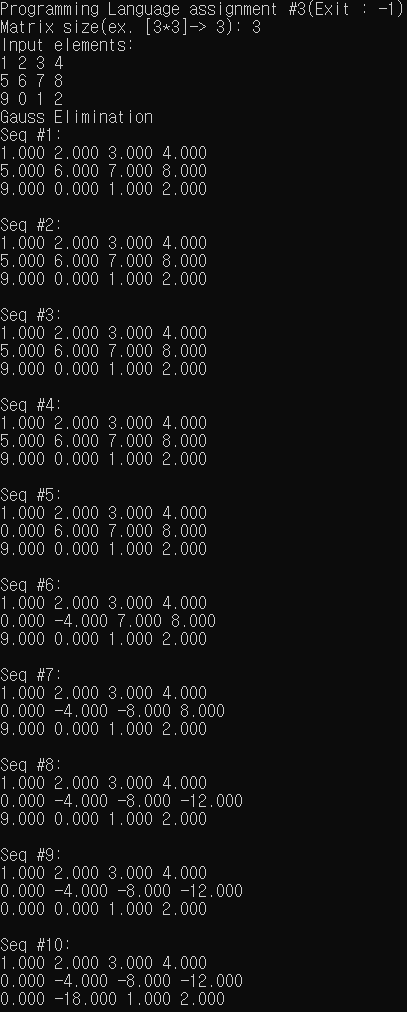
1. 7장 온라인 강의를 다 들은후 7장 연습문제중 (pp.350-351)    9,10,11,13,19,20 번 문제를 푸시오
2. N 값을 입력으로하여 N 개의 소수(prime number)를 출력하는 C언어 프로그램을 작성하고,  N이 각각 100, 1000, 5000, 10000 일때 프로그램의 실행시간을 측정하여 보시오

|  |
| --- |
| import java.util.Scanner;  public class gaussjordan {  private static double[][] matrix;  private static Scanner scanner = new Scanner(System.in);  public static void main(String[] args) {  int size;  System.out.println("Programming Language assignment #3(Exit : -1)");  System.out.println("Matrix size(ex. [3\*3]-> 3): ");  size = scanner.nextInt();  matrix = creatMatrix(size);  assignMatrix(size);  gauss\_jordan(size);  printMatrix(size);  }  private static double[][] creatMatrix(int size) {  return new double[size][size + 1];  }  private static void assignMatrix(int size) {  int i, j;  System.out.println("Input element:");  for(i = 0; i < size; i++)  for(j = 0; j < size + 1; j++)  matrix[i][j] = scanner.nextDouble();  }  private static void printMatrix(int size) {  int i, j;  for (i = 0; i < size; i++) {  for (j = 0; j < size + 1; j++)  System.out.print(matrix[i][j] + " ");  System.out.println();  }  }  private static void swap(int size, int a, int b) {  int i;  for (i = 0; i < size + 1; i++) {  double temp = matrix[a][i];  matrix[a][i] = matrix[b][i];  matrix[b][i] = temp;  }  }  private static void gauss\_jordan(int size) {  int i, j, k;  // 가우스 소거  for (i = 0; i < size - 1; i++) {  // 행렬 대각선이 0일 경우 행 swap  if (matrix[i][i] == 0) {  for (j = i + 1; j < size; j++)  if (matrix[j][i] != 0) {  swap(size, i, j);  break;  }  i--;  // 행렬이 대각선이 아닌 경우  } else {  double divide = matrix[i][i]; //연산을 하면서 i번째 계수가 바뀌므로 미리 저장해놓아야합니다  for (j = 0; j < size + 1; j++)  matrix[i][j] \*= (1 / divide); //i번째 행에 R(i)의 i번째 계수의 역수를 곱합니다  for (j = i + 1; j < size; j++){  double multiply = matrix[j][i]; //마찬가지로 matrix[j][i]도 연산을 하면서 값이 바뀌므로 미리 저장해놓아야합니다  //multiply가 필요한 이유: i번째 계수를 1로 만들기 위해  for (k = 0; k < size + 1; k++)  matrix[j][k] -= (matrix[i][k] \* multiply); //R(i)번째 행과 R(j)번째 행을 더합니다  }  }  }  double temp = matrix[size - 1][size - 1]; //마지막 행의 i번째 계수를 저장해놓습니다  for (i = 0; i < size + 1; i++)  matrix[size - 1][i] \*= (1 / temp); //마지막 행에 i번째 계수의 역수만큼 곱합니다  // 가우스 조던 소거  for (i = size - 1; i > 0; i--) { // 거꾸로  for (j = i - 1; j >= 0; j--) {  double multiply = matrix[j][i]; //연산하면서 바뀌는 값이니 미리 저장  for (k = i; k < size + 1; k++)  matrix[j][k] -= (matrix[i][k] \* multiply);  }  }  }  } |

[코드 2] gaussjordan.java

* 실행 결과
  + C: gaussjordan.c



* + JAVA: gausjordan.java

