**1. array[12][42] 같은 배열을 지역 혹은 전역변수로 선언하지 말고 malloc() 같은 함수를 이용하여 달력을 저장하기 위한 배열의 크기만큼 heap에 할당한 후에 이곳에 포인터를 이용하여 참조 혹은 값을 저장하는 명령을 사용하도록 합니다. 포인터 변수를 사용할 때는 p+i와 같이 포인터를 한번에 바꾸지 않고, 반드시 p++같이 하나씩만 증가 혹은 감소하는 명령을 사용하여 실행속도를 향상하도록 합니다. 프로그램의 구조나 논리는 이전 프로그램과 같음.**

소스코드 링크: <https://github.com/Codejune/C_ProgrammingLanguage/tree/master/C_GaussJordan>

|  |
| --- |
| #pragma warning(disable:4996)  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  #include <string.h>  double\*\* matrix;  void creatMatrix(int size) { // 행렬생성  int i;  matrix = (double\*\*)calloc(size, sizeof(double\*));  for (i = 0; i < size; i++)  matrix[i] = (double\*)calloc(size + 1, sizeof(double));  }  void assignMatrix(int size) { // 행렬입력  int i, j;  printf("input Elements:\n");  for (i = 0; i < size; i++) {  for (j = 0; j < size + 1; j++)  scanf("%lf", &matrix[i][j]);  }  void printMatrix(int size) { // 행렬출력  int i, j;  for (i = 0; i < size; i++) {  for (j = 0; j < size + 1; j++)  printf("%.3lf ", matrix[i][j]);  printf("\n");  }  }  void swap(int size, int a, int b) { // 행교환  for (int i = 0; i < size + 1; i++) {  double temp = matrix[a][i];  matrix[a][i] = matrix[b][i];  matrix[b][i] = temp;  }  }  void gauss\_jordan(double\*\* matrix, int size) { // 행렬연산  // 가우스 소거  for (int i = 0; i < size - 1; i++) {  // 행렬의 대각선이 0일 경우  if (!matrix[i][i]) {  for (int j = i + 1; j < size; j++)  if (matrix[j][i]) {  swap(matrix, size, i, j); // 행 swap  break;  }  i--;  // 행렬의 대각선이 0이 아닐 경우  } else {  double divide = matrix[i][i]; // 대각선 계수 원본 저장    for (int j = 0; j < size + 1; j++)  matrix[i][j] \*= (1 / divide); // 역수를 취해 1을 만듦    for (int j = i + 1; j < size; j++){  double multiply = matrix[j][i];  for (int k = 0; k < size + 1; k++)  matrix[j][k] -= (matrix[i][k] \* multiply); // 남은 연산 처리    }  }  }  double temp = matrix[size - 1][size - 1];  for (int i = 0; i < size + 1; i++)  matrix[size - 1][i] \*= (1 / temp);  // 가우스 조던 소거  for (int i = size - 1; i > 0; i--) { // upper소거  for (int j = i - 1; j >= 0; j--) {  double multiply = matrix[j][i];  for (int k = i; k < size + 1; k++)  matrix[j][k] -= (matrix[i][k] \* multiply);  }  }  }  int main(void) {  int size;  printf("Programming Language assignment #3(Exit : -1)\n");  printf("Matrix size(ex. [3\*3]-> 3): ");  scanf("%d", &size);  creatMatrix(size);  assignMatrix(size);  gauss\_jordan(matrix, size);  printMatrix(size);  } |

[코드 1] gaussjordan.c

실행 결과





 