Advanced C Programming – Homework #1

Prof. Cheolsoo Park

Assistant: Heesang Eom

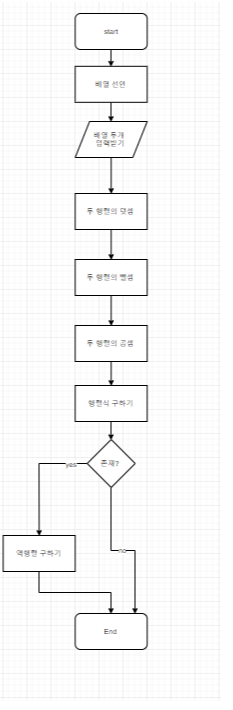
컴퓨터정보공학부 2019202050 이강현

Submit date : 2019 / 10 / 15

문제 5

1.과제 소개: 두 행렬을 입력받고 두 행렬의 덧셈 뺄셈 곱셈을 구하고 역행렬이 존재하는지 알고 존재한다면 역행렬을 구하는 예제이다.

2.설계 내용: 두 행렬을 입력받고 행렬의 연산정의에 의해 배열을 이용하여 연산을 하고 역행렬의 존재는 행렬식의 값이 0이 아님을 조건문으로 달아 역행렬을 구하는 코드를 짜려고 한다.



3. 코드 및 주석

#include <stdio.h>//헤더 파일 선언

int main()//main 함수 선언

{

int arr1[3][3] = { 0 };//배열 선언 및 초기화

int arr2[3][3] = { 0 };//배열 선언 및 초기화

int arr3[3][3] = { 0 };//배열 선언 및 초기화

int i = 0, j = 0;//변수 선언

int num1 = 0, num2 = 0;//변수 선언

int mul = 0;//변수 선언

int det = 0;//변수 선언

for (i = 0; i < 3; i++)//입력받는 반복문

{

for (j = 0; j < 3; j++)

{

printf("arr1배열 입력 받기: ");

scanf("%d", &arr1[i][j]);

}

}

for (i = 0; i < 3; i++)//입력받는 반복문

{

for (j = 0; j < 3; j++)

{

printf("arr2배열 입력 받기: ");

scanf("%d", &arr2[i][j]);

}

}

printf("\n");

/\*두 행렬의 덧셈\*/

printf("덧셈\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

printf("%d ", arr1[i][j] + arr2[i][j]);

}

printf("\n");//개행

}

printf("\n");//개행

/\*두 행렬의 뺄셈\*/

printf("뺄셈\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

printf("%d ", arr1[i][j] - arr2[i][j]);

}

printf("\n");//개행

}

printf("\n");//개행

/\*두 행렬의 곱셈\*/

printf("곱셈\n");

for (i = 0; i < 3; i++)

{

for (int k = 0; k < 3; k++)

{

mul = 0;

for (j = 0; j < 3; j++)

mul += arr1[i][j] \* arr2[j][k];

printf("%d ", mul);

}

printf("\n");//개행

}

/\*행렬식 구하기\*/

for (i = 0; i < 3; i++) //3번 반복 for문

det += ((arr1[0][i] \* arr1[1][(i + 1) % 3] \* arr1[2][(i + 2) % 3]) - (arr1[0][i] \* arr1[1][(i + 2) % 3] \* arr1[2][(i + 1) % 3])); // 행렬식 계산

if (det == 0)// det=0 일 경우( 역행렬의 존재성 판정 )

printf("\n 역행렬 존재하지 않음 \n"); // 문자열 출력

else

{ // det=0이 아닐 경우

printf("\n 역행렬 \n"); // 문자열 출력

for (j = 0; j < 3; j++) // 세번 반복 for문

{

for (i = 0; i < 3; i++)

{ // 세번 반복 for문

printf("%g\t", ((arr1[(i + 1) % 3][(j + 1) % 3] \* arr1[(i + 2) % 3][(j + 2) % 3]) - (arr1[(i + 2) % 3][(j + 1) % 3] \* arr1[(i + 1) % 3][(j + 2) % 3]))); // 역행렬 출력

}

printf(")\n"); // 문자열 출력

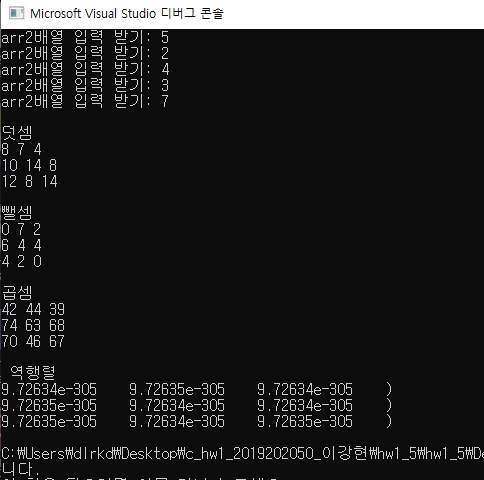
}

} // 역행렬 계산

return 0;//종료

}

4. 실행 결과



5. 고찰: 역행렬을 구하는 알고리즘이 굉장히 어려웠다. 역행렬을 구하는 방법은 알고 있었지만 이를 배열에 접근하여 코딩을 하려다 보니 좀처럼 되지 않았다. 결국에는 해내지 못했지만 코딩을 하는 과정에는 수학적인 사고력과 지식이 있어야 함을 깨닫게 되었다.