아주대학교 프로그래밍언어

201821048 미디어학과 이서영

2019/11/25 함수, 배열, 정렬

# 소스코드 및 주석

/\*

[과제#8] 배열과 정렬을 이용한 성적처리 프로그램

요구사항 : 위 실습 내용을 참고해서 5명의 학생 점수에 해당하는 임의의 점수를

0~99점 사이로 생성하고 이에 대한 각 항목별 환산점수로 표시해서

각 학생별 총점과 순위를 구하는 프로그램을 작성하시오.

출력형태 : 각자 나름대로 자유로운 형식으로 출력 가능함.

(단, 점수 표시는 소수점 첫째자리까지만)

화면에 출력되는 형태는 위의 표의 형태로 나오거나

아니면, 난수에 의해서 발생되는 점수만 2차원 배열의 형태로 출력이 되고

학생별로 총점과 순위를 표시해도 됩니다.

추가 기능 구현 :

1) 최대값 및 최소값 계산 및 표시

2)평균값 표시

순서 :

//학번부여, 0으로 초기설정//전체 표 출력

//랜덤으로 점수 부여//전체 표 출력

//가중치 계산//전체 표 출력

//순위 계산//전체 표 출력

//최솟값 출력

//최댓값 출력

//평균 출력

핵심내용 :

배열을 돌면서 랜덤한 값을 각각의 자리에 넣는다.

배열을 이용해 값을 전달한다.

배열을 돌면서 비교해 가장 큰 값과 작은 값을 찾는다.

배열을 돌면서 원소를 더하고 그 개수로 나누어 평균을 구한다.

for문을 이용해 배열의 조건을 통제한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*여기까지 과제내용\*\*\*\*\*\*\*\*\*

추가 점수!!! : 원하는 기준으로 순서대로 정렬해 보기 편하도록 구현하기

//위치를 바꿀 학생 1,2 넣어 내용 교환

//원하는 기준에 따라 정렬해 출력

이 두 함수를 만들어 모든 기준에 따라 정렬이 되는 것을 확인했다.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define row 8

#define column 8

void print\_array(double student[][column]) {

//전체 표 출력

printf("학번\*\*\*\*\*출석\*\*\*\*\*중간\*\*\*\*\*기말\*\*\*\*\*과제\*\*\*\*\*수시\*\*\*\*\*총점\*\*\*\*\*순위\*\*\*\*\n");

// 0 1 2 3 4 5 6 7

for (int i = 0; i < row - 3; i++) {

//0-5 세로 출력

for (int j = 0; j < column; j++) {

if (j == column - 8 || j == column - 1) {

printf("|%7.0lf|", student[i][j]);

}

else {

printf("|%7.1lf|", student[i][j]);

}

}

printf("\n");

}

printf("\n");

}

void set\_zero\_array(double student[][column]) {

//학번부여, 0으로 초기설정

for (int i = 0; i < row + 1; i++) {

//0-8 세로 0으로 설정

student[i][0] = (double)i + 1;

for (int j = 1; j < column; j++) {

//1-7 가로 0으로 설정

student[i][j] = 0;

}

}

printf("학번부여, 0으로 초기설정\n");

print\_array(student);

}

void set\_random\_array(double student[][column]) {

//랜덤으로 점수 부여

for (int i = 0; i < row - 3; i++) {

//0-4 세로 랜덤으로 설정

for (int j = 1; j < column - 2; j++) {

//1-5 가로 랜덤으로 설정

student[i][j] = rand() % 100;

}

}

printf("랜덤으로 점수 부여하기\n");

print\_array(student);

}

void weight\_get\_finalscore(double student[][column]) {

//가중치 계산

double atten = 0.1, mid = 0.2, fin = 0.2;

double hw = 0.3, atti = 0.2;

for (int i = 0; i < row; i++) {

student[i][1] = student[i][1] \* atten;

}

for (int i = 0; i < row; i++) {

student[i][2] = student[i][2] \* mid;

}

for (int i = 0; i < row; i++) {

student[i][3] = student[i][3] \* fin;

}

for (int i = 0; i < row; i++) {

student[i][4] = student[i][4] \* hw;

}

for (int i = 0; i < row; i++) {

student[i][5] = student[i][5] \* atti;

}

for (int i = 0; i < row; i++) {

double sum = 0;

for (int j = 1; j < column - 2; j++) {

sum = sum + student[i][j];

}

student[i][6] = sum;

}

printf("가중치 계산 후\n");

print\_array(student);

}

void mark\_ranking(double student[][column]) {

//순위 계산

for (int rank = 1; rank < row - 2; rank++) {

int rank\_flag = row - 2;

for (int i = 0; i < row; i++) {

if (student[i][6] > student[rank\_flag][6] && student[i][7] < 1) {

rank\_flag = i;

}

}

student[rank\_flag][7] = rank;

}

printf("순위 계산 후\n");

print\_array(student);

}

void minimum(double student[][column]) {

//최솟값 출력

printf("min : ");

for (int i = 1; i < column; i++)

{

student[row][i] = 1000;

int flag = row;

for (int j = 0; j < 5; j++) {

if (student[j][i] < student[flag][i]) {

flag = j;

}

}

student[5][i] = student[flag][i];

}

for (int j = 1; j < column - 1; j++) {

printf("|%7.1lf|", student[5][j]);

}

printf("\n");

}

void maximum(double student[][column]) {

//최댓값 출력

printf("max : ");

for (int i = 1; i < column; i++)

{

student[row][i] = -1;

int flag = row;

for (int j = 0; j < 5; j++) {

if (student[j][i] > student[flag][i]) {

flag = j;

}

}

student[6][i] = student[flag][i];

}

for (int j = 1; j < column - 1; j++) {

printf("|%7.1lf|", student[6][j]);

}

printf("\n");

}

void average(double student[][column]) {

//평균 출력

printf("average :");

for (int i = 1; i < column; i++) {

double average = 0;

for (int j = 0; j < 5; j++) {

average = average + student[j][i];

}

average = average / 5;

student[7][i] = average;

}

for (int j = 1; j < column - 1; j++) {

printf("|%7.1lf|", student[7][j]);

}

printf("\n");

}

void switch\_value(double student[][column], int fir\_stu, int sec\_stu) {

//위치를 바꿀 학생 1,2 넣어 내용 바꿈

for (int i = 0; i < column; i++) {

double temp;

temp = student[fir\_stu][i];

student[fir\_stu][i] = student[sec\_stu][i];

student[sec\_stu][i] = temp;

}

}

void bubble\_sort\_ranking(double student[][column], int sort\_column) {

//원하는 기준에 따라 정렬해 출력하기

for (int o = row - 4; o > 0; o--) {

for (int i = 0; i < o; i++) {

if (student[i][sort\_column] < student[i + 1][sort\_column]) {

switch\_value(student, i, i + 1);

}

}

}

print\_array(student);

}

int main() {

srand((unsigned)time(NULL));

double student[row + 1][column];

set\_zero\_array(student);

//학번부여, 0으로 초기설정//전체 표 출력

set\_random\_array(student);

//랜덤으로 점수 부여//전체 표 출력

weight\_get\_finalscore(student);

//가중치 계산//전체 표 출력

mark\_ranking(student);

//순위 계산//전체 표 출력

minimum(student);

//최솟값 출력

maximum(student);

//최댓값 출력

average(student);

//평균 출력

printf("\n");

printf("출석점수 순서로 정렬");

printf("\n");

bubble\_sort\_ranking(student, 1);

printf("중간점수 순서로 정렬");

printf("\n");

bubble\_sort\_ranking(student, 2);

printf("기말점수 순서로 정렬");

printf("\n");

bubble\_sort\_ranking(student, 3);

printf("과제점수 순서로 정렬");

printf("\n");

bubble\_sort\_ranking(student, 4);

printf("수시평가점수 순서로 정렬");

printf("\n");

bubble\_sort\_ranking(student, 5);

printf("총점점수 순서로 정렬");

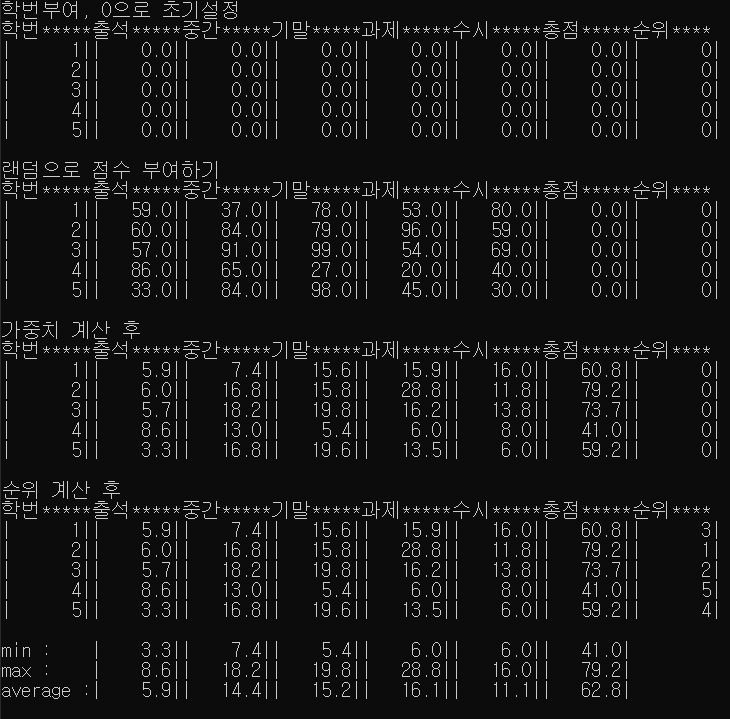
printf("\n");

bubble\_sort\_ranking(student, 6);

return 0;

}

# 실행결과



과제를 하면서 배열과 for문을 함께 사용하는 능력이 향상된 것을 느꼈다.

정렬을 하면서 수업시간에 배운 정렬 외에도 다른 방법을 찾아봤고 구현이 간단한 것을 이용해 추가 과제를 하였다.

