/\*------------------------------------------------------------------------------------------------

C-Programming Asignment 3-3 : 만 명의 성적의 평균, 분산 등 분포도를 알려주는 프로그램.

학번 : B611019

전공 : 컴퓨터공학과 2학년

이름 : 김남일

날짜 : 2019년 05월 21일

주요 내용 :

프로그램의 목적은 10000명의 점수를 나누고 그들의 평균, 분산, 표준편차, 그리고 A+, A의 점수컷과 인원수를 알아내는 것이다. 처음에 배열 len[7], People[10000], score[10000]을 선언한다. People배열에 만명의 사람을 각각 7개 그룹으로 나눈다.(상위 2퍼, 7퍼...) 그리고 나서 각 그룹에 사람이 들어갈 때 마다, len[i]에 사람수를 넣는다. (예를 들어 len[0]은 1번째, len[1]은 2번째 그룹의 사람수이다.get\_len함수로 구현.)그룹 구분이 완료되면, score배열에 그룹마다 10점의 점수분포로 나누어 점수를 랜덤으로 할당해준다. 예를 들어 아까 get\_len함수를 통해 할당된 그룹중 첫번째 그룹(상위 2퍼) 1그룹은 최저 85점, 최고 94점으로 점수를 넣어준다. 똑같이 바로 아래 그룹은 최저 75점, 최고 84...을 반복해 최저 25점, 최고 34점인 그룹까지 점수를 넣어준다. 이때 주의할 점은 아까 get\_len울 통해서 구한len[i]를 이용해야 한다는 점이다. 각 그룹은 len[i](그룹에 속한 사람의 수)만큼의 점수를 넣어줘야 하므로, for (i = 0; i < len[1]; i++) score[len[0] + i] = (rand() % 10) + 75 참조할 때,

score의 대괄호[]를 주목해보자. len[0]+i가 처음 시작 인덱스 인것은 처음의 1그룹사람들의 마지막 번호 다음 번호부터 시작해서 len[1](2번째 그룹의 사람수)까지 점수가 넣어주기 위함이다. 그 다음 그룹의 "시작 인덱스"에 score[len[0] +len[1] + i]처럼 그 전의 사람들 수를 넣어준다. (get\_score함수로 구현)

점수 할당이 끝났으면 score[0]부터 score[9999]까지 모두 정렬한다. 내림차순으로 score[0]은 25점, score[9999]는 94점이다.(sort함수로 구현) 평균은 10000명의 점수의 합을 10000으로 나눠주고, 분산은 ((score[i]-평균)^2)/10000)로 구한다. 표준편차는 분산에 sqrt를 붙여주면 된다. A+의 점수컷은 score[8999]의 점수(1000등)를 기준으로 잡았다. 그리고 그 점수보다 같거나 위인 사람들이 A+이다. (대략 상위10퍼) A의 점수컷은 score[6999]의 점수(3000등)를 기준으로 잡았고, 이 점수보다 같거나 위면서, A+컷 점수보다 낮은 사람들이 A이다.(대략 상위 30퍼까지) 각각 get\_mean, get\_var, A1\_grade, A2\_grade함수를 통해서 구현했다.

-----------------------------------------------------------------------------------------------\*/

#include <stdio.h>

#include <time.h>

#include <math.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

#define SIZE 10000

void sort(int score[]) { // [0]에서 부터 [9999]까지 순서대로 내림차순으로 정렬해주는 함수.

int i, j, temp, least;

for (i = 0; i < 10000; i++)

{

least = i;

for (j = i + 1; j < 10000; j++)

if (score[j] < score[least])

least = j;

temp = score[i];

score[i] = score[least];

score[least] = temp;

}

}

void get\_len(int People[] ,int len[]){

int i;

srand((unsigned)time(NULL));

for (i = 0; i < SIZE; i++) // len배열을 이용해서 각각의 그룹별 사람수를 구해준다.

{

People[i] = rand() % 100;

if (98 <= People[i] && People[i] <100) //상위 2퍼

++len[0];

if (93 <= People[i] && People[i] <98) // 상위 7퍼

++len[1];

if (78 <= People[i] && People[i] <93) //상위 22퍼

len[2]++;

if (23 <= People[i] && People[i] <78) // 상위 78퍼

len[3]++;

if (8 <= People[i] && People[i] <23) // 상위 93퍼

len[4]++;

if (2 <= People[i] && People[i] <8) // 상위 98퍼

len[5]++;

if (0 <= People[i] && People[i] < 2) // 상위 100퍼

len[6]++;

}

}

void get\_score(int score[], const int len[]) { // 속해있는 그룹마다 점수를 차례대로 넣어준다. 그룹순서가 빠를수록 높은점수이다(단, 그룹내에선 무작위).

int i;

srand((unsigned)time(NULL));

for (i = 0; i < len[0]; i++)

score[i] = (rand() % 10) + 85;

for (i = 0; i < len[1]; i++)

score[len[0] + i] = (rand() % 10) + 75;

for (i = 0; i < len[2]; i++)

score[len[0] +len[1] + i] = (rand() % 10) + 65;

for (i = 0; i < len[3]; i++)

score[len[0] + len[1]+ len[2] + i] = rand() % 10 + 55;

for (i = 0; i < len[4]; i++)

score[len[0] + len[1] + len[2]+len[3] + i] = rand() % 10 + 45;

for (i = 0; i < len[5]; i++)

score[len[0] + len[1] + len[2] + len[3]+len[4] + i] = rand() % 10 + 35;

for (i = 0; i < len[6]; i++)

score[len[0] + len[1] + len[2] + len[3] + len[4]+ len[5] + i] = rand() % 10 + 25;

}

double get\_mean(int score[]){ //평균을 구해주는 함수

int i;

double sum = 0, mean = 0;

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

sum += score[i];

mean = (sum / SIZE);

return mean;

}

double get\_var(int score[]){ // 분산을. 구해주는 함수

int i;

double var=0;

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

var +=(score[i] - get\_mean(score))\*(score[i] - get\_mean(score));

return var/SIZE;

}

int A1\_grade(int score[]){ // 상위 10%인 사람의 점수를 기준으로 그 점수보다 같거나 위이면 A+이다.

int i, index = 0;

int A1\_cut;

A1\_cut = score[8999];

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

if (score[i] >= A1\_cut)

index++;

return index;

}

int A2\_grade(int score[]){ // 상위 30%인 사람의 점수를 기준으로 그 점수보다 같거나 위이고, A+점수보다 낮으면 A이다.

int i, index = 0;

int A2\_cut;

A2\_cut = score[6999];

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

if (score[i] >= A2\_cut)

index++;

return index - A1\_grade(score);

}

int main() {

int i;

int len[7] = { 0 }; //각 그룹의 인원수를 저장해준다.

int People[10000] = {0};// 사람들 10000명의 정규분포 곡선을 형성해준다.

int score[10000] = {0}; // 나눠진 사람들의 점수가 각각 들어간다.

get\_len(People, len);

get\_score(score, len);

sort(score);

printf("만 명의 그룹별 사람수: (오른쪽으로 갈수록 점수가 높아진다.)\n");

for ( i = 6; i >=0 ;i--)

printf("%d ", len[i]);

printf("\n");

printf("평균은: %f \n", get\_mean(score));

printf("분산은: %f \n", get\_var(score));

printf("표준편차는: %f \n", sqrt(get\_var(score)));

printf("A+의 점수컷 :%d \n" ,score[8999]);

printf("A+의 인원수 :%d \n",A1\_grade(score));

printf("A의 점수컷 :%d \n", score[7999]);

printf("A의 인원수 :%d \n", A2\_grade(score));

return 0;

}