/\*

240103 알고리즘 1번 문제

배열에 n개의 성적데이터를 읽어 평균을 구하고, 평균과 평균 이상인 데이터의 수를 출력

입력 : n, scores(array)

결과 : avg, over\_count

1) 배열에 읽으면서 합을 구한다.(반복) - sum

2) 평균을 구한다 - avg

3) 평균에 저장된 데이터를 보면서 평균보다 큰 데이터의 수를 구한다 - count

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define DNUM 100

void main()

{

int scores[DNUM], n, i, sum, over\_count;

double avg;

over\_count = 0;

sum = 0;

printf("The number of data to be..");

scanf("%d",&n);

printf("Enter %d data : \n",n);

for (i = 0; i < n; i++) {

printf("Enter %d data : \n", i);

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

sum = (sum + scores[i]);

}

avg = (double)sum / n;

for (i = 0; i < n; i++) {

if (scores[i] > avg) over\_count = over\_count + 1;

}

printf("Average = %.2f\n", avg);

printf("over\_count = %d\n", over\_count);

}



/\*

240103 알고리즘 2번 문제

음수가 나올 때 까지 양수를 읽어 들여 읽어 들인 데이터 수와 최대 값을 출력하는 C 프로그램을 작성하시오

단, 처음부터 음수를 입력하는 경우 "no data"라고 출력하고 종료

입력 : num

결과 : size,max

1) 초기치 설정, max=0으로 시작, 데이터의 수 size는 0으로 초기화

2) 시뮬레이션- 데이터 num은 1개 읽어 양수인 경우 처리 과정 시작

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

void main()

{

int num, max, size, number;

size = 0;

max = 0;

printf("숫자를 입력해주세요 \n");

scanf("%d",&num);

if (num < 0) {

printf("No Data!");

return 1;

}

printf("숫자를 입력해주세요. 다 입력하셨으면 음수를 입력하여 종료하십시오 :\n");

while (num > 0) {

scanf(" %d", &num);

size = size + 1;

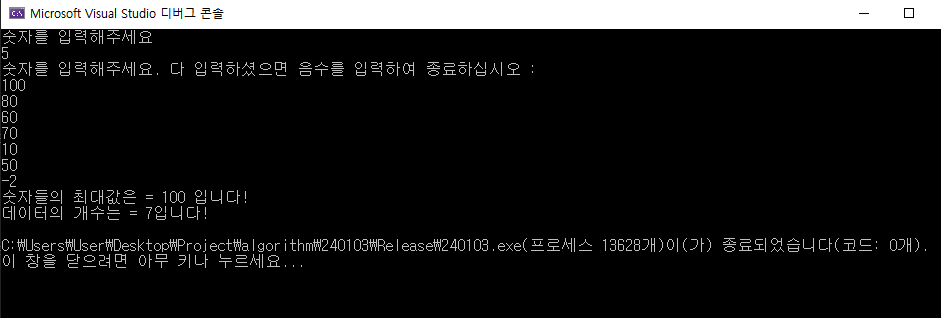
if (max < num) max = num;

}

printf("숫자들의 최대값은 = %d 입니다!\n", max);

printf("데이터의 개수는 = %d입니다!\n", size);

}



/\*

240103 알고리즘 3번 문제

n명의 학생의 학번과 성적을 이차원 배열에 읽어 들여 학번을 입력하면 그의 성적과 등수를 출력하는 C프로그램을 작성하시오.

입력 : n, scores(n,2), sid , scores는 n \* 2 이차원 배열

결과 : jumsu, rank

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

void main()

{

int i, scores[100][2], sid, jumsu, rank, n;

printf("Enter n :");

scanf("%d", &n);

printf("Input id and score : \n");

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

scanf("%d %d", &scores[i][0], &scores[i][1]);

printf("Input student id to be search:");

scanf("%d", &sid);

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

if (sid == scores[i][0]) {

jumsu = scores[i][1];

break;

}

rank = 0;

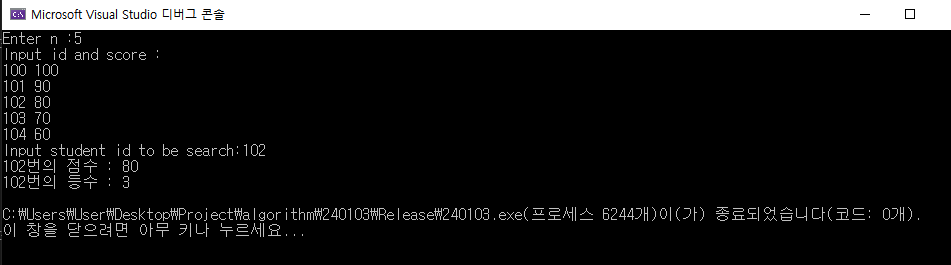
for (i = 0; i < n; i++)

if (scores[i][1] > jumsu) rank++;

printf("%d번의 점수 : %d\n", sid, jumsu);

printf("%d번의 등수 : %d\n", sid, rank+1);

}



/\*

240103 알고리즘 4번 문제

C함수 작성법 - FOR 문

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int my\_pow(int x, int y);

void main()

{

int k;

for (k = 2; k < 6; k++)

printf("%d \*\* %d = %d\n", k, k + 1, my\_pow(k, k + 1));

}

int my\_pow(int x, int y) {

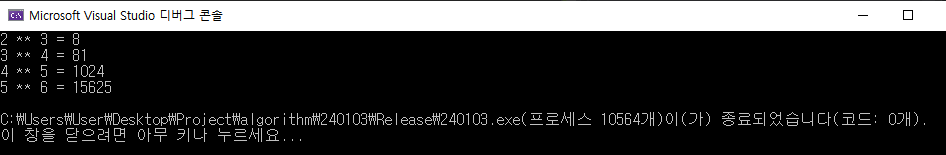
int i, ans = 1;

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

ans = ans \* x;

return ans;

}



/\*

240103 알고리즘 5번 문제

배열을 파라메터로 넘겨주기

(매우중요) 배열 이름은 배열의 첫 데이터의 주소 -> data = &data[0]

int pcount(int a[])

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int pcount(int a[])

{

int j, result = 0;

for (j = 0; j < 10; j++)

if (a[j] > 0) result++;

return result;

}

void main()

{

int data[10], k;

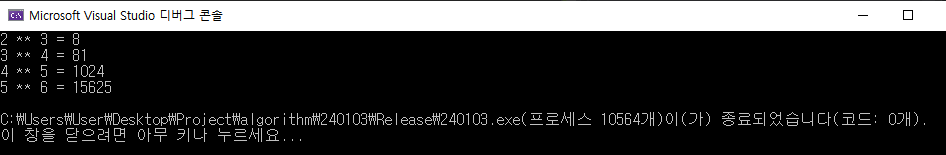
printf("Enter 10 data: ");

for (k = 0; k < 10; k++)

scanf("%d", &data[k]);

printf("양의 갯수는 %d\n", pcount(data));

}



/\*

240103 알고리즘 6번 문제

주어진 main함수에 대해 함수를 추가해보기

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define DNUM 100

int pcount(int a[], int n);

double avg(int a[], int n);

int max(int a[], int n);

int min(int a[], int n);

void main()

{

int data[DNUM], k, n;

printf("처리할 데이터의 수 : ");

scanf("%d", &n);

printf("%d개의 정수를 입력하세요. ", n);

for (k = 0; k < n; k++)

scanf("%d", &data[k]);

printf("양의 갯수는 %d\n", pcount(data, n));

printf("데이터 평균= %.2f\n", avg(data, n));

printf("최대값 = %d\n", max(data, n));

printf("최소값 = %d\n", min(data, n));

}

double avg(int a[], int n)

{

int sum = 0;

double iavg = 0;

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

sum = sum + a[i];

}

iavg = sum / n;

return iavg;

}

int min(int a[], int n)

{

int imin = a[0], i;

for (i = 1; i < n; i++) {

if (a[i] < imin) imin = a[i];

}

return imin;

}

int max(int a[], int n)

{

int imax = a[0], i;

for (i = 1; i < n; i++) {

if (a[i] > imax) imax = a[i];

}

return imax;

}

int pcount(int a[], int n)

{

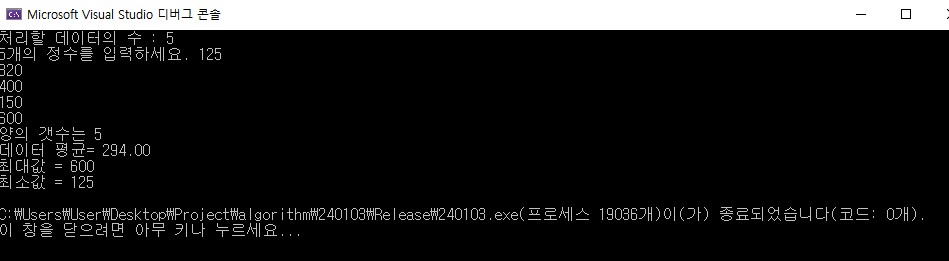
int j, result = 0;

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

if (a[j] > 0) result++;

return result;

}



/\*

240103 알고리즘 7번 문제

n명의 학생의 학번과 0~100 점사이의 학생성적을 이차원 배열에 읽어 들여 성적의 평균과 최고점을 출력,

학번을 입력하면 그의 등수를 출력하는 알고리즘을 작성하라

알고리즘은 4가지 제어 구조로 명령어를 구성한다

1. 순차적 제어
2. 선택적 제어
3. 반복적 제어 -> 반복제어를 exit 할 수 있을 지 검토
4. 함수 프로그램의 호출과 복귀

1) 성적의 평균을 구하는 함수 find\_avg 작성

2) 성적의 최고점을 구하는 함수 find\_max 적성

3) 학번을 가지고 등수를 구하는 함수 find\_rank 작성

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define DNUM 100

double find\_avg(int a[], int n);

int find\_max(int a[], int n);

int find\_rank(int a[], int n);

void main()

{

int data[DNUM][2], k, sid, n;

printf("학생들의 수를 입력하세요 : ");

scanf("%d", &n);

printf("학생들의 학번과 성적을 입력하세요. ", n);

for (k = 0; k < n; k++)

scanf("%d %d", &data[k][0], &data[k][1]);

printf("최고성적 = %d\n", find\_max(data, n));

printf("평균= %.2f\n", find\_avg(data, n));

printf("검색하고자 하는 학생의 학번을 입력하세요. ", n);

scanf("%d", &sid);

printf("%d학번 학생의 등수는 %d\n", find\_rank(data, n));

}

double find\_avg(int a[][], int n)

{

int sum = 0;

double iavg = 0;

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

sum = sum + a[i][1];

}

iavg = sum / n;

return iavg;

}

int find\_rank(int a[], int n)

{

int rank = 0;

for (int i = 0; i < n; i++)

if (a[i] > a[i]) rank+1;

}

int find\_max(int a[], int n)

{

int imax = a[0], i;

for (i = 1; i < n; i++) {

if (a[i] > imax) imax = a[i];

}

return imax;

}

/\*

240103 알고리즘 8번 문제

재귀기법(recursive) : 어떤 것을 정의할 때 자신의 부분으로 재정의 하는 것(ex. factorial!)

트리구조 : 스택을 이용하여 반복으로 구현할 수 있지만 복잡하다

유클리드 알고리즘 : 최대공약수 구하기

gcd 함수 응용

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define DNUM 100

void main()

{

int result, result2;

int a, b, cond = 1;

while (cond) {

printf("최대공약수를 구할 두 수를 입력하세요. \n");

scanf("%d %d", &a, &b);

result = gcd(a, b);

result2 = gcd2(a, b);

printf("두 수의 최대공약수는 %d입니다 \n", result);

printf("두 수의 최대공약수는 %d입니다 \n", result2);

printf("계속하실래요?(1/0)\n");

scanf("%d", &cond);

}

}

int gcd(int x, int y) // 재귀적 정의법

{

if (y == 0)

return x;

if (x == 0)

return y;

else

return gcd(y, (x % y));

}

int gcd2(int x, int y) // 반복을 이용한 구현

{

int tmp;

while (y != 0) {

tmp = y;

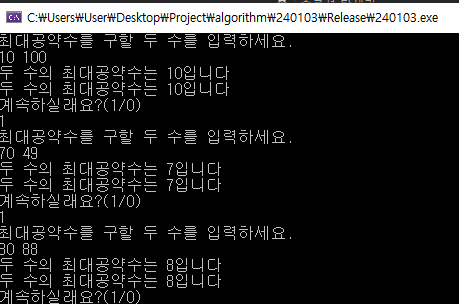
y = x % y;

x = tmp;

}

return x;

}



/\*

240103 알고리즘 9번 문제

main의 명령인수 받아들이기

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

void main(int argc, char\* argv[])

{

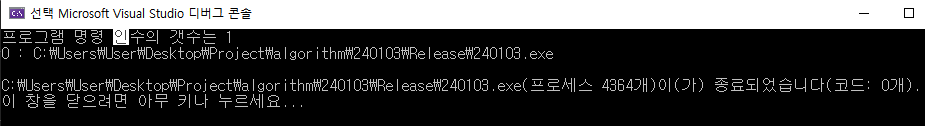
int i;

printf("프로그램 명령 인수의 갯수는 %d\n", argc);

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

printf("%d : %s\n", i, argv[i]);

}



/\*

240103 알고리즘 10번 문제

파일 확인

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

void main(int argc, char\* argv[])

{

int a, b, result, answer, correct = 0, i = 0;

FILE\* infile;

FILE\* out;

if (argc != 3) {

printf("실행인수를 잘못 주었습니다...\n");

exit(1);

}

if ((infile = fopen(argv[1], "r")) == NULL) {

printf("입력 파일을 열 수 없습니다. \n");

exit(1);

}

if ((out = fopen(argv[2], "w")) == NULL) {

printf("출력 파일을 만들 수 없습니다. \n");

exit(1);

}

while (fscanf(infile, "%d %d", &a, &b) != EOF) {

printf("%d : gcd(&d, &d) = ", i + 1, a, b);

scanf("%d", &answer);

result = gcd(a, b);

if (answer != result)

{

printf("Answer : gcd(%d, %d) = %d \n",a,b,result);

fprintf(out, "%d %d \n", a, b);

}

else {

printf("Correct...\n");

correct++;

}

i++;

}

printf("%d 문제 중에 %d 문제 통과하셨습니다...\n,i,correct");

}

int gcd(int x, int y) // 재귀적 정의법

{

if (y == 0)

return x;

if (x == 0)

return y;

else

return gcd(y, (x % y));

}