/\*

240105 알고리즘 1번 문제 - 구조체

const의 값은 언제 결정되는가? 컴파일하면서 결정됨

변수는 Run할때 정해짐

define보다 const가 더 명확하게 오류를 잡을 수 있도록 해준다.

구조체 : 여러 형의 데이터를 하나의 객체로 선언하여 사용함

exmember중에 성적이 4.0 이상인 학생의 id를 출력하시오.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int count = 0;

// 구조체 샘플

/\*

struct studentTag {

long id; //int

char name[10];

char grade;

int score; //foat

};

typedef struct studentTag2 {

long id2;

char name[10];

char grade;

int score;

}student;

typedef struct member { //이것으로 member라는 구조체 형이 선언, tpyedef에 의하여 example이라는 구조체로 정의됨

int id;

char name[20];

float score;

}example;

\*//////////////////////////////////////////////////////////////////////////////////////////////////

typedef struct member {

int id;

char name[20];

float score;

}example;

void over40(example\* a, int n) {

int k;

printf("List of high score students(over 4.0)\n");

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

if (a[k].score >= 4.0) {

printf("%d -- %s\n", a[k].id, a[k].name);

}

};

example find\_max(example\* a, int n) {

int k, max\_index;

max\_index = 0;

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

if (a[k].score > a[max\_index].score)

max\_index = k;

return a[max\_index];

};

void main()

{

example onep, exmember[20];

int i, n;

printf("input the number of data : ");

scanf(" %d", &n);

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

printf("데이터 입력 :");

scanf("%d %s %f", &exmember[i].id, &exmember[i].name, &exmember[i].score);

if (exmember[i].score >= 4.0) {

count = count + 1;

}

}

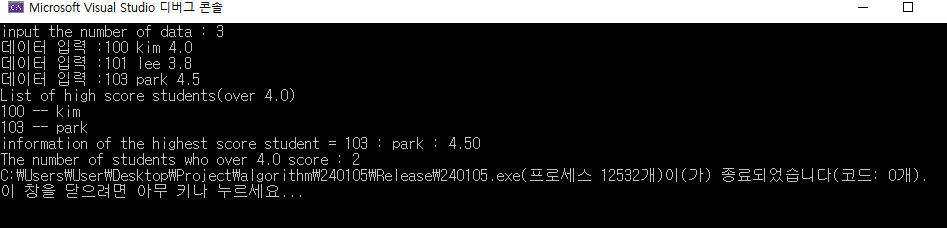
over40(exmember, n);

onep = find\_max(exmember, n);

printf("information of the highest score student = %d : %s : %.2f\n", onep.id, onep.name, onep.score);

printf("The number of students who over 4.0 score : %d", count);

}



/\*

240105 알고리즘 2번 문제 - 마방진(magic square)

이차원 배열을이해하고 정해진 규칙을 프로그램 코드로 바꾸는 연습

문제 정의

n\*n 행렬로서 각 행의 합, 열의 합, 두 대각선의 합이 같도록 1부터 n^2의 값을 가지는 경우

n이 3인 경우(합 15)

6 | 1 | 8

7 | 5 | 3

2 | 9 | 4

첫번째 행의 중앙에 1을 위치(5-5 마방진)

15 | 8 | 1 | 24 | 17 |

16 | 14 | 7 | 5 | 23 |

22 | 20 | 13 | 6 | 4 |

3 | 21 | 19 | 12 | 10 |

9 | 2 | 25 | 18 | 11 |

1. 시작은 1,3 또는 0,2에서 진행

2. 기본적으로 오른쪽에서 왼쪽 대각선 방향으로 왼쪽에서 오른쪽 대각선 방향으로 행은 감소, 열은 증가

시켜가면서 숫자를 채운다.

3. 행 감소 열 증가를 수행하다가 행이1(0)보다 작으면 행은 1이 된다.

4. 행감소 열 증가를 수행하면서 위치를 결정하다가 열이 5보다 크면 열이 1이 된다.

5. 채우는 숫자가 5의 배수의 다음 숫자일 경우 채울 위치는 이전 위치에서 행만 1증가한다.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int count = 0;

// 마방진 - 0으로 초기화된 n\*n의 2차원 배열 table[n][n], 출력은 n\*n의 마방진

/\*

행을 움직이는 변수 i, 열을 움직이는 변수 j

133p

첫번째 행의 중앙에 1을 놓는다 - i=0, j=(n-1)/2 table[i][j] = 1

왼쪽 대각선 방향으로 올라가면서 빈자리에 1씩 큰 수를 위치

이때 행렬의 밖으로 벗어나면 그 방향의 반대편에서 계속

이동하려는 자리에 숫자가 있을 경우, 이동을 시작하는 지점 바로 아래에 숫자를 위치한다.

알고리즘

for count = 2 to n=n

row - i-1, col = j-1

if (row < 0) row = n-1

if (col < 0) col = n-1

if (table[row][col] !=0) i++

else i=row, j= col

이차원 배열 table을 모두 0으로 초기화하는 이유는 무엇인가?

다음 두개의 함수로 분리하여 다시 작성하여 실행해보자

void make\_msquare(int table[][MAX\_SIZE], int n)

void display(int table[][MAX\_SIZE], int n)

\*//////////////////////////////////////////////////////////////////////////////////////////////////

#define MAX\_SIZE 15

void main()

{

int n, r, c, sum = 0, digit, row, col;

int table[MAX\_SIZE][MAX\_SIZE];

printf("마방진 제작 프로그램입니다\n");

printf("마방진의 행렬 개수를 입력해주세요.\n");

scanf("%d", &n);

if ((n < 1) || n > MAX\_SIZE) {

printf("오류 발생, 마방진의 크기가 너무 큽니다.\n");

exit(0);

}

if (!(n % 2)) {

printf("오류 발생, 홀수를 입력해주시기 바랍니다.\n");

exit(0);

}

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

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

table[r][c] = 0;

r = 0; c = (n - 1) / 2;

table[r][c] = 1;

for (digit = 2; digit <= n \* n; digit++){

row = r - 1;

col = c - 1;

if (row < 0) row = n - 1; //up

if (col < 0) col = n - 1; //left

else {

r = row;

c = col;

}

table[r][c] = digit;

}

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

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

printf("%5d", table[r][c]);

printf("\n");

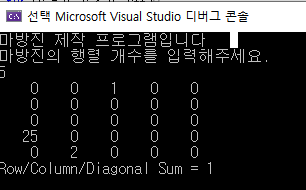
}

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

sum += table[0][c];

printf("Row/Column/Diagonal Sum = %d\n", sum);

} // main



/\*

240105 알고리즘 3번 문제 - rane, srand

0 ~ RAND\_MAX - 1 정수값을 난수로 발생시켜 리턴 해 주는 함수

문제점 : 같은 난수 발생

해결책 : 실행도중 seed값을 달리해주어야 실행할 때 마다 같은 난수 발생 방지

seed값을 설정해주는 srand()함수

0~100 사이의 난수를 발생하여 출력하는 c프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

int main()

{

int i, num;

srand((unsigned int)time(NULL));

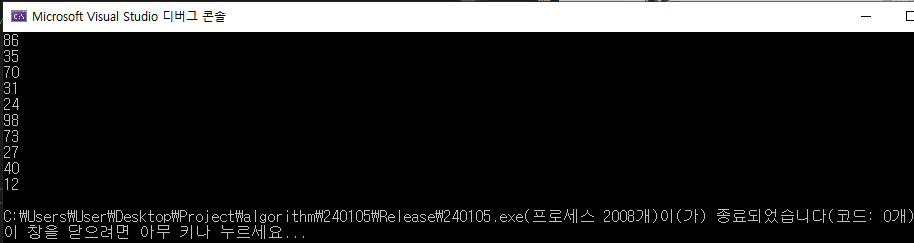
for (i = 0; i < 10; i++) {

num = rand() % 101;

printf("%d\n", num);

}

} // main



/\*

240105 알고리즘 4번 문제 - rane, srand

0 ~ RAND\_MAX - 1 정수값을 난수로 발생시켜 리턴 해 주는 함수

문제점 : 같은 난수 발생

해결책 : 실행도중 seed값을 달리해주어야 실행할 때 마다 같은 난수 발생 방지

seed값을 설정해주는 srand()함수

a~z까지의 문자 난수 생성 프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX 26

//////////////////////////////////////////////////////////////////////////////////////////////////

int main()

{

char ch[MAX];

int i, j;

srand((unsigned int)time(NULL));

ch[0] = 'a' + rand() % 26;

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

ch[i] = 'a' + rand() % 26;

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

if (ch[i] == ch[j]) {

i = j;

break;

}

}

}

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

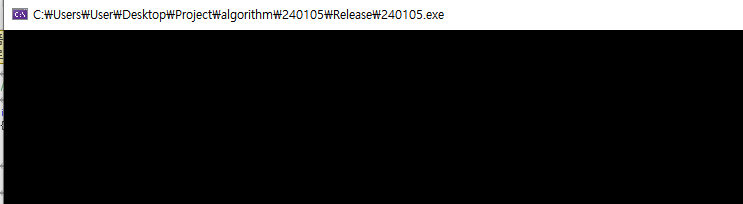
printf("%c", ch[i]);

}

printf("\n");

system("pause");

} // main



/\*

240105 알고리즘 4번 문제 - rane, srand

0 ~ RAND\_MAX - 1 정수값을 난수로 발생시켜 리턴 해 주는 함수

문제점 : 같은 난수 발생

해결책 : 실행도중 seed값을 달리해주어야 실행할 때 마다 같은 난수 발생 방지

seed값을 설정해주는 srand()함수

a~z까지의 문자 난수 생성 프로그램

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define NUM 20

//////////////////////////////////////////////////////////////////////////////////////////////////

int main()

{

float rdata[NUM], fsum = 0.0;

int i, isum = 0, temp;

srand((unsigned int)time(NULL)); // seed값 설정

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

rdata[i] = rand() % 100 / 100.0; // 난수를 생성해 rdata[]에 저장,

0 ~ (99 + .0.99)

printf("%.2f ", rdata[i]);

if ((i + 1) % 7 == 0) { //줄 바꾸기

printf("\n");

}

}

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

temp = rdata[i];

isum += temp;

fsum += (rdata[i] - temp);

}

printf("\n\n정수부분의 합 : %d\n", isum);

printf("\n소수 부분의 합 %.3f\n", fsum);

system("pause");

} // main

0 ~ (99 + .0.99) 링크 오류

/\*

240105 알고리즘 5번 문제 - lookup table

입력

n\*2 pdb(code:price)

오늘 판매한 물품의 개수(num)

물품코드(item\_code),판매개수(item\_num)

출력 : 가격의 합계

구현해야할 함수코드 : display\_code(pdb,n)

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX\_NUM 50

//////////////////////////////////////////////////////////////////////////////////////////////////

long lookup(long pdb[][2], int n, long item\_code);

long display\_code(long pdb[][2], int n);

long lookup(long pdb[][2], int n, long item\_code) {

int i;

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

if (pdb[i][0] == item\_code) return pdb[i][1];

}

return 0;

}

long display\_code(long pdb[][2], int n) {

int i;

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

printf("&ld &ld\n", pdb[i][0], pdb[i][1]);

}

return 0;

}

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

FILE\* pricedb; // 물품코드 가격

long pdb[MAX\_NUM][2], item\_code, item\_price;

int i = 0, j, num, n, item\_num;

long total\_price = 0;

//데이터 파일 읽어오기

/\*

}

while (fscanf(pricedb, "%ld %ld", &(pdb[i][0]), &(pdb[i][1])) != 0

i++;

}

\*/

n = i; //물품의 총 갯수

printf("오늘 팔린 물품의 종류는 몇 개인가요? : ");

scanf("%d", &num);

display\_code(pdb, n);

for (j = 0; j < num; j++) {

printf("물품코드와 팔린 갯수를 입력하시오. ");

scanf("%id %id", &item\_code, &item\_num);

item\_price = lookup(pdb, n, item\_code);

if (item\_price == 0) {

printf("%ld : 입력한 물품에 대한 가격 정보가 없습니다. ", item\_code);

}

else {

printf("%ld : %ld\n", item\_code, item\_price);

}

total\_price = total\_price + (item\_num \* item\_price);

}

printf("오늘의 정산 금액 : %ld\n", total\_price);

system("pause");

} // main