

**과제 2 조건반복문**



|  |  |
| --- | --- |
| **과 목 :** | **기초컴퓨터프로그래밍** |
| **제출 일자 :** | **2018. 5. 06** |
| **담당 교수 :** | **김승태** |
| **학 과 :** | **사회복지학과, 통계학과** |
| **학 번 :** | **20170299, 20165389** |
| **이 름 :** | **김유빈, 정다희** |

**과제 방식**

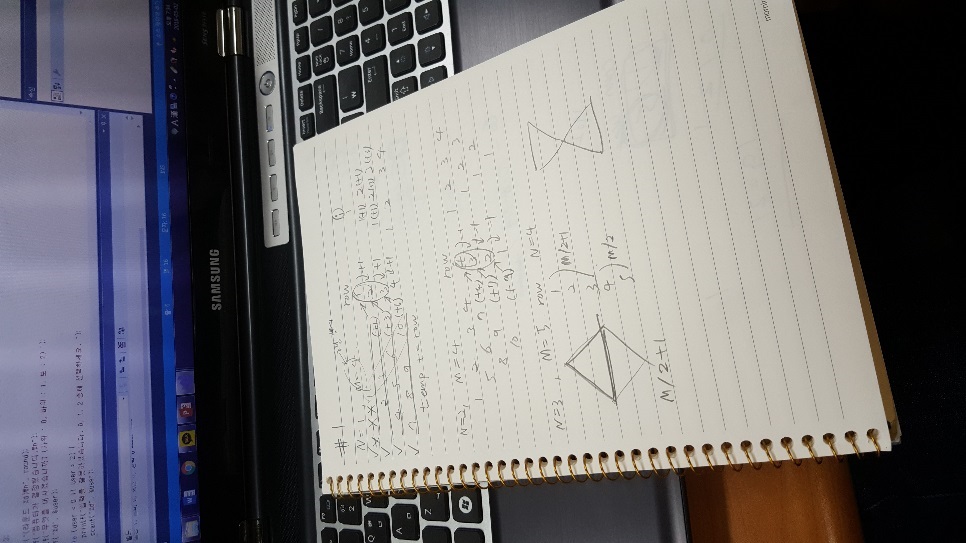
: 아바타는 ‘김유빈’학생이, 이 학생에게 명령하는 사람은 ‘정다희’학생이 했다. 우리는 5월 2일 수요일에 수업이 끝나자마자 강의실에 남아서 과제의 절반 이상을 끝냈다.

 수요일에 끝내지 못한 과제는 그 다음날인 5월 3일에 만나서 끝냈다.

5월 2일 수요일 1 5월 3일 목요일

이 노트는 5월 2일에 필기한 노트이다.

제일 어려운 첫번째 과제를 고민한 흔적이다.



**과제 2 – 1 숫자 피라미드**

N에 따라 모양이 다르므로 switch 문을 이용하여 case 별로 나누었다.

해결해야할 핵심 포인트는 크게 두가지였다.

1. 예를 들어 N=1일때, 첫째 줄에 숫자 1개, 둘째 줄에 2개, 셋째 줄에 3개…처럼 줄에 따라 숫자의 개수를 조절하는 것
2. 삼각형, 역삼각형, 다이아몬드, 모래시계처럼 모양을 만드는 것

1번에서 가장 어려웠던 점은 다음줄로 넘어 갈때마다 숫자가 다시 1로 돌아가지 않고 그대로 +1씩 더해서 숫자가 바꾸어 지게 하는 점이다. 이를 위해 temp라는 새로운 변수를 만들어 다음 줄로 넘어갈 때마다 1씩 증가하게 했다.

2번을 해결하는게 이 과제의 핵심 포인트였다. 처음에는 등차수열을 이용하여 풀려고 했으나 복잡하여 넘어갔다. 우리는 ‘빈칸’에 초점을 맞추어 줄에 따라 줄어드는 빈칸 수에 착안하여 새로운 규칙을 발견하였다. 변수 gap이라는 새로운 변수를 만들었다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

int N, M, row, gap, i, temp = 0;

printf("N : ");

scanf("%d", &N);

printf("M : ");

scanf("%d", &M);

while ((N == 3 || N == 4) && M % 2 == 0) { //N=3,4일 때 M의 값이 홀수가 아닐 때 다시 입력받도록 하는 장치

printf("N 값이 3이나 4일 경우에는 M 값은 홀수만 가능합니다. M 값을 다시 입력하세요.\n");

printf("M : ");

scanf("%d", &M);

}

switch (N)

{

case 1: for (row = 1; row <= M; row++) { //'줄'에 따라 숫자의 개수와 빈칸을 조절

for (gap = 0; gap < M - row; gap++) //예를 들어 M=4일 때, 첫째 줄 빈칸이 3칸,둘째 줄 빈칸 2칸,셋째 줄 빈칸 1칸, 넷째 줄 빈칸 0칸이다.

printf(" ");

//이때 빈칸 규칙을 발견하여 줄 마다 M-row까지 빈칸을 만듦

for (i = 1; i <= row; i++)

//첫째 줄에 숫자1, 둘째 줄에 숫자 2,3, 셋째 줄에 숫자 4,5,6, ... 를 출력

printf("%4d", i + temp);

//이때 %4d를 한 이유는 M값이 커질수록 숫자가 겹치지 않게 하기 위해서. “%d “로 하면 교수님 예시처럼 예쁜 삼각형이 나오나 M값이 커질수록(M은 최대20) 삼각형 모양이 이상함

temp += row;

//다음 줄로 넘어가도 다시 숫자1로 돌아가지 않고 그대로 1씩 더해지도록 하는 장치 (변수 temp)

printf("\n");

//첫째 줄에 숫자1개, 둘째 줄에 숫자2개...씩으로 줄에 따른 숫자의 갯수를 조절함

}

break;

case 2: for (row = M; row > 0; row--) { //N=1일 때와 반대로 만들기 위해 row를 M부터 시작하여 1개씩 줄여서 센다

//나머지는 N=1일 때와 동일

for (gap = 0; gap < M - row; gap++)

printf(" ");

for (i = 1; i <= row; i++)

printf("%4d", i + temp);

temp += row;

printf("\n");

}

break;

case 3: for (row = 1; row <= M / 2 + 1; row++) { //다이아몬드는 N=1과 N=2를 합침

//먼저 N=1일 때를 출력. 단, 여기서 규칙을 발견하여 M/2+1개 줄까지만 출력.

for (gap = 0; gap < M / 2 + 1 - row; gap++)

printf(" ");

for (i = 1; i <= row; i++)

printf("%4d", i + temp);

temp += row;

printf("\n");

}

for (row = M / 2; row > 0; row--) { //N=2일 때를 출력. 이때는 M/2개의 줄만 출력. M/2부터 1개의 줄을 출력

for (gap = 0; gap <= M / 2 - row; gap++)

printf(" ");

for (i = 1; i <= row; i++)

printf("%4d", i + temp); //temp변수의 값은 N=1일때의 값을 그대로 이어 받으므로 숫자는 계속 증가함

temp += row;

printf("\n");

}

break;

case 4: for (row = M / 2 + 1; row > 0; row--) { //역삼각형은 N=2와 N=1을 합침

//N=3일때와 비슷하게 N=2를 먼저 출력하는데 M/2+1부터 1까지 줄을 센다

for (gap = 0; gap <= M / 2 + 1 - row; gap++)

printf(" ");

for (i = 1; i <= row; i++)

printf("%4d", i + temp);

temp += row;

printf("\n");

}

for (row = 2; row <= M / 2 + 1; row++) { //N=1을 출력. 주의 점은 모래시계에 가운데 숫자는 이미 N=2일때 출력해서 row=2부터 시작

for (gap = 0; gap < M / 2 + 1 - row; gap++)

printf(" ");

for (i = 1; i <= row; i++)

printf("%4d", i + temp);

temp += row;

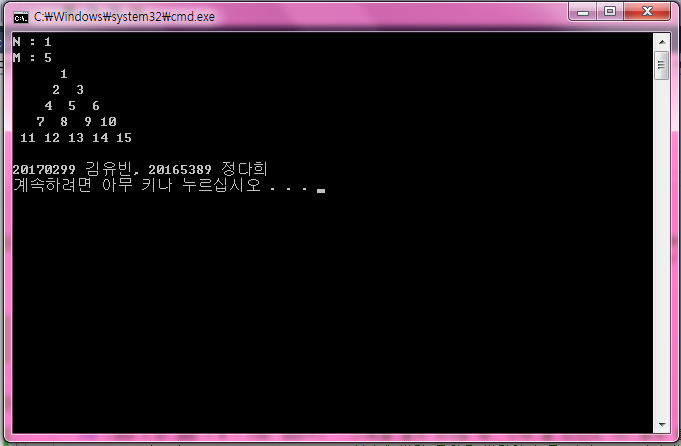
printf("\n");

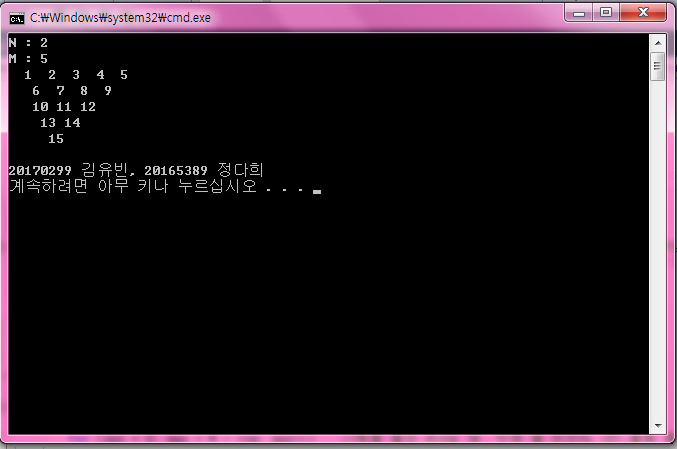
}

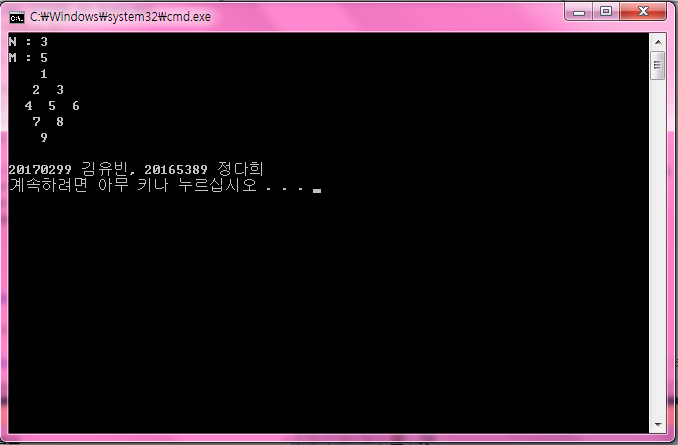
}

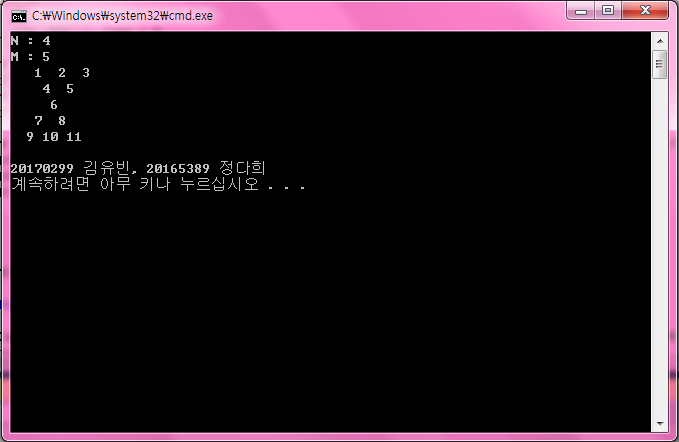
printf("\n20170299 김유빈, 20165389 정다희\n");

}









**과제 2 – 2 구구단**

과제 2의 가장 핵심 포인트는 한 가지이다.

바로 단의 개수에 따라 단마다의 빈칸을 조절하는 것이다.

처음에는 빈칸의 수를 조절하는 것에만 초점을 두었다. 각 단의 왼쪽에 빈칸을 두는 것은 떠올릴 수 있었다. 그러나 예시와 최대한 비슷하게 하기 위해 일정한 직사각형 안에 구구단을 모두 출력한다는 점을 고려하여 for문을 수정하였다.

참고로 변수 j는 한 단마다 9개의 곱셈을 출력하는 장치이며 변수 i는 구구단을 몇 개까지 출력하는지를 조절하는 장치이다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

int N, M, i, j, gap;

scanf("%d %d", &N, &M);

for (j = 1; j < 10; j++) {

for (i = N; i <= M; i++) {

for (gap = 0; gap < (80 - 10 \* (M - N + 1)) / (M - N + 2); gap++)

printf(" ");

printf("%d \* %d = %2d", i, j, i\*j);

}

printf("\n");

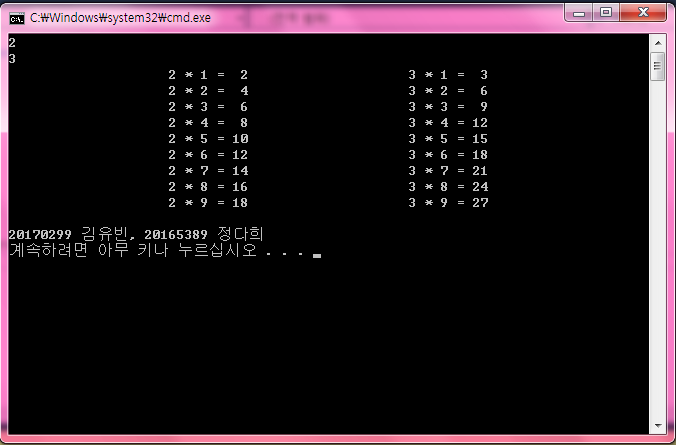
}

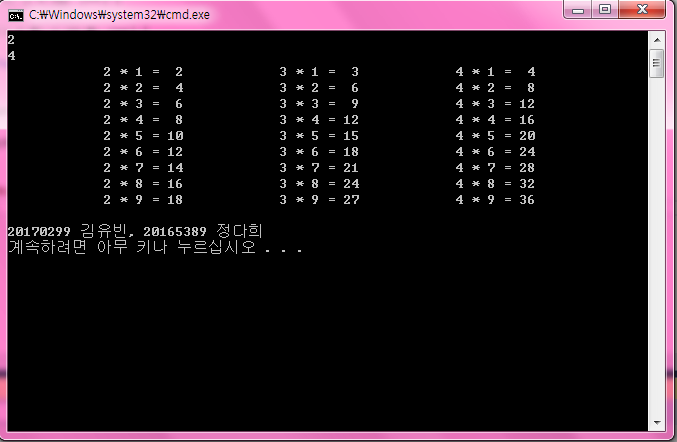
printf("\n20170299 김유빈, 20165389 정다희\n");

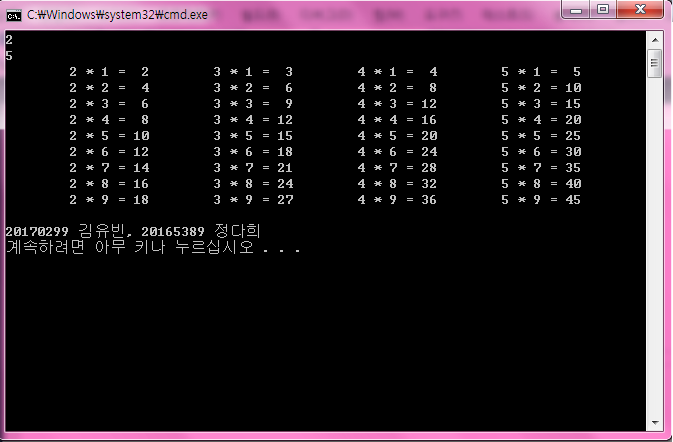
}

for (gap = 0; gap < (80 - 10 \* (M - N + 1)) / (M - N + 2); gap++) 에서 ‘80’은 대략 커다란 직사각형의 가로 길이이다. 예를 들어 ‘2 \* 9 = 18’의 길이가 대략 10칸이다. 최대 4개의 단을 출력할 수 있으며 1개의 단마다 1개의 빈칸이 있으므로 80으로 길이를 정했다. 전체 길이 80에 단이 차지하는 길이(10 \* (M - N + 1))를 빼고 나머지 빈칸을 동일하게 나누기위해 (M - N + 2)로 분할했다.

여기서 주의할 점은 한 단을 세로로 완성하여 다음 단으로 넘어간다고 생각하지 말고 첫째 줄 모두를 완성하고 다음 줄로 넘어가는 것으로 알아야 한다. 예를 들어, 2 \* 9 = 18 3 \* 9 = 27 과 같이 가로줄을 완성한다는 방식이다.







**과제 2 – 3 요일 찾기**

쉬운 해결책은 년(年)을 계산하고 그 다음 월(月), 마지막으로 일(日)을 계산하여 전체 일수를 통해 요일을 구하는 것이다.

이때의 킬링 포인트는 월(月)을 계산할 때 윤년의 2월을 어떻게 다루냐는 것이다.

처음에는 변수 yoon를 만들어 yoon=1일 때 윤년이고 yoon=0일 때 윤년이 아님을 나타내려고 했으나 년(年)계산할 때는 아무 문제가 없는데 그대로 yoon에 값이 들어간 상태에서 입력한 해의 2월이 윤년인지 아닌지를 계산하는데 문제가 생겨서 변수yoon를 사용하지 않았다. 대신 if문을 사용하여 해당 해가 윤년인지 아닌지에 따라 2월의 일수를 달리 했다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main() {

int year, month, day, totalday, i; //변수 totalday가 1900년 1월 1일부터 입력날짜까지 며칠이 지났는지를 계산

printf("yyyy mm dd ? ");

scanf("%d %d %d", &year, &month, &day);

totalday = day - 1; //먼저 날짜를 totalday에 넣어주는데 1900년1월1일부터 세므로 여기에 1을 빼기

for (i = 1900; i < year; i++) {

//윤년을 계산하여 윤년일 때 366일, 윤년이 아닐 때 365일을 더한다

//단, 입력한 해의 바로 전까지만 계산

if (i % 4 != 0 || ((i % 100 == 0) && (i % 400 != 0)))

totalday += 365;

else

totalday += 366;

}

switch (month - 1) { //입력한 월을 계산

case 11: totalday += 30;

case 10: totalday += 31;

case 9: totalday += 30;

case 8: totalday += 31;

case 7: totalday += 31;

case 6: totalday += 30;

case 5: totalday += 31;

case 4: totalday += 30;

case 3: totalday += 31;

case 2: totalday += 28;

case 1: totalday += 31;

}

if (((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) && month > 2)

totalday++; //입력한 해가 윤년이면 2월은 29일이므로 totalday에 1을 더한다

switch (totalday % 7) { //마지막은 전체 일수를 더한 totalday를 통해 입력한 날짜의 요일을 계산

case 0: printf("월요일입니다.\n");

break;

case 1: printf("화요일입니다.\n");

break;

case 2: printf("수요일입니다.\n");

break;

case 3: printf("목요일입니다.\n");

break;

case 4: printf("금요일입니다.\n");

break;

case 5: printf("토요일입니다.\n");

break;

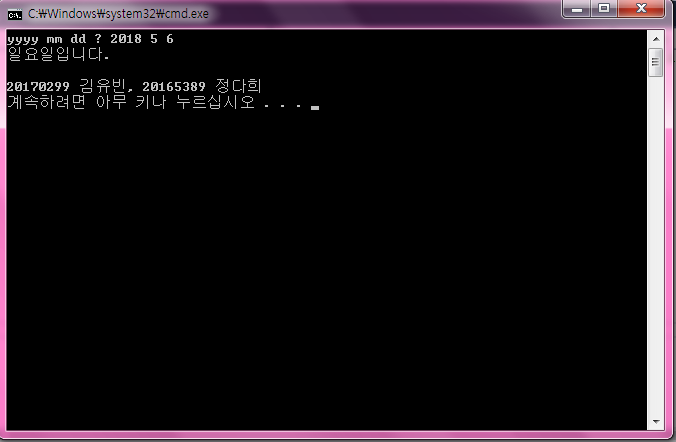
case 6: printf("일요일입니다.\n");

break;

}

printf("\n20170299 김유빈, 20165389 정다희\n");

}



**과제 2 – 4 가위바위보 게임**

가위바위보의 가장 힘든 점은 라운드마다 누가 이겼는지 카운트를 새고 그에 따라 경기를 종료시켜야 한다는 점이다. 처음에는 변수 com\_win만 사용하여 컴퓨터가 3승할 때까지 카운트 하려고 했으나 그러면 사용자가 3승하는 것을 판단하지 못한다. 따라서 컴퓨터가 3승을 할 때까지 동시에 컴퓨터가 3번 질 때까지 카운트가 계속되도록 했다.

참고로 컴퓨터가 가위,바위,보 중에 무엇을 냈는지에 따라 다시 if문을 사용하여 사용자의 가위바위보를 통해 누가 이겼는지를 판단하도록 하였다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define GAWUI 0

#define BAWUI 1

#define BO 2

int main(void) {

int round, com, user, com\_win = 0, com\_lose = 0;

//컴퓨터 이긴것과 컴퓨터 진것 즉, 사용자가 이긴 것과 관련된 변수

printf("가위바위보 게임\n컴퓨터 : 0승 0패, 당신 : 0승 0패\n\n");

for (round = 1;(com\_win < 3 && com\_lose < 3);round++) {

//컴퓨터가 3승하기 전까지 라운드를 카운트

//단, 컴퓨터 이긴 것만을 계산하면 사용자가 3승 넘게 이겨도 컴퓨터가 3승할 때까지 계산하므로 컴퓨터가 3승 진것 즉, 사용자가 이긴 것도 계산해야 함

srand(time(NULL));

com = rand();

com %= 3;

printf("(라운드 %d)\n", round);

printf("컴퓨터가 결정했습니다.\n");

printf("무엇을 내시겠습니까? (가위 : 0, 바위 : 1, 보 : 2) ");

scanf("%d", &user); //사용자가 가위바위보를 냄

while (user < 0 || user > 2) { //사용자가 가위바위보 이외를 입력하는 것을 방지하기 위해

printf("입력을 잘못하셨습니다. 0, 1, 2 중에 선택하세요. ");

scanf("%d", &user);

}

switch (com) {

//여기서 누가 이겼는지 계산. 컴퓨터가 무엇을 냈는지에 따라 case분류

case GAWUI: if (user == BAWUI) {

printf("컴퓨터는 가위, 당신은 바위, 당신이 이겼습니다.\n");

com\_lose++; //사용자 이긴 것 카운트

}

else if (user == BO) {

printf("컴퓨터는 가위, 당신은 보, 컴퓨터가 이겼습니다.\n");

com\_win++; //컴퓨터 이긴 것 카운트

}

else printf("컴퓨터도 가위, 당신도 가위, 비겼습니다.\n");

break;

case BAWUI: if (user == BO) {

printf("컴퓨터는 바위, 당신은 보, 당신이 이겼습니다.\n");

com\_lose++;

}

else if (user == GAWUI) {

printf("컴퓨터는 바위, 당신은 가위, 컴퓨터가 이겼습니다.\n");

com\_win++;

}

else printf("컴퓨터도 바위, 당신도 바위, 비겼습니다.\n");

break;

case BO: if (user == GAWUI) {

printf("컴퓨터는 보, 당신은 가위, 당신이 이겼습니다.\n");

com\_lose++;

}

else if (user == BAWUI) {

printf("컴퓨터는 보, 당신은 바위, 컴퓨터가 이겼습니다.\n");

com\_win++;

}

else printf("컴퓨터도 보, 당신도 보, 비겼습니다.\n");

break;

}

printf("컴퓨터 : %d승 %d패, 당신 : %d승 %d패\n\n", com\_win, com\_lose, com\_lose, com\_win); //결론 도출

}

if (com\_win == 3) printf("컴퓨터가 이겼습니다\n");

else printf("당신이 이겼습니다.\n");

printf("\n20170299 김유빈, 20165389 정다희\n");

}

