**Introduction to Computer and Lab**

**Homework #8**

**Due date: Jun 5, 2016**

**학번: 201404051**

**이름: 정 용 석**

**1. 두 집합의 합집합 출력**

**1.1 Solution**

**크기 20의 랜덤 수 집합 2개를 생성하고, 앞에서부터 한 개씩 원소의 합을 더해가고, 이를 출력한다.**

**1.2. Source code**

void findUnionSet() {

//2개의 합집합의 합 출력

int set1[SIZE];

int set2[SIZE];

int i, j, check;

int sum = 0;

//generateRandomSet함수와 동일

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

{

while (1) {

set1[i] = rand() % 100 + 1;

set2[i] = rand() % 100 + 1;

check = 0;

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

if ((set1[j] == set1[i]) || (set2[j] == set2[i])) {

check = 1;

break;

}

}

if (!check)

break;

}

}

//배열의 첫번째 원소들로부터 한개 씩 더한다.

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

sum += set1[i] + set2[i];

//출력

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

printf("%d ", set1[i]);

printf("\n");

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

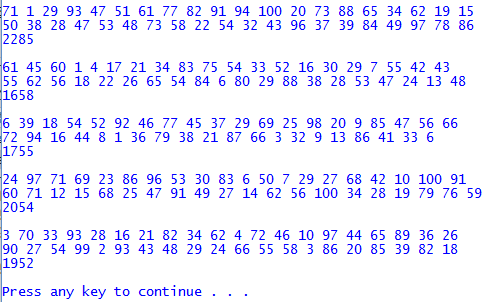
printf("%d ", set2[i]);

printf("\n");

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

}

**1.3. Result (snapshot)**



**2 Histogram 출력**

**2.1. Solution**

**크기가 1000인 배열 안에 1부터 20사이의 값을 가지는 난수를 생성하고, 이중 반복 문을 통하여 각 배열의 index+1값이 각각 몇 번 나오는지 확인하고 크기 20의 배열에 각각 빈도 수를 저장한다. 각각의 빈도 수만큼 반복 문을 돌려서 ‘\*’를 출력하면 된다.**

**2.2. Source code**

void printHistogram(){

//Histogram 출력

int frequency = 0; //빈도수

int freqSet[20];

int randNumber[1000];

int i, j;

int sum = 0;

//1000개의 난수 생성 후 배열에 삽입

for (i = 0; i < 1000; i++)

randNumber[i] = rand() % 20 + 1;

//1부터 20까지 난수 배열과 비교하여 빈도 수 계싼

for (i = 0; i < 20; i++)

{

for (j = 0; j < 1000; j ++)

if (randNumber[j] == i + 1) {

frequency++;

}

//빈도 수 계산 후 빈도 수 배열에 삽입

freqSet[i] = frequency;

frequency = 0;

}

//각 숫자별 빈도 수 \*로 출력

for (i = 0; i < 20; i++) {

printf("%d ", i + 1);

for (j = 0; j < freqSet[i]; j++) {

printf("\*");

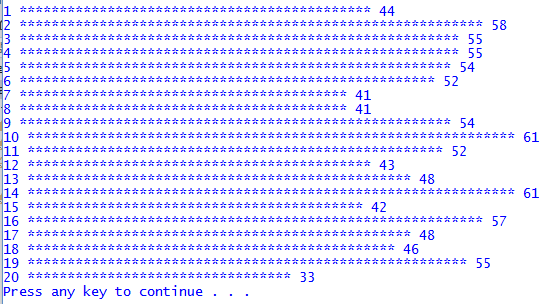
}

printf(" %d\n", freqSet[i]);

}

}

**2.3. Result (snapshot)**



**3 배열을 이용한 성적표 출력**

**3.1. Solution**

**학생 수가 5명이기 때문에 크기가 5인 배열과, 각 학생 당 과목을 저장 할 수 있도록 4 X 5의 2차원 배열을 만들고 사용자로부터 각각 입력을 받는다. 그리고 각 ID와 과목 출력과 동시에 총점과 평균을 계산하여 함께 출력한다.**

**3.2. Source code**

void printScoreReport(){

//배열을 이용한 성적표 출력

int sum = 0, average; //총합, 평균

int ID[5]; //ID 5개

int score[4][5]; //과목 4개 X 학생 수 5명

int i, j;

//배열의 첫 행에 ID 값을 저장

//배열의 각 열에 각 과목에 대한 점수 저장

for (i = 0; i < 5; i++) {

printf("Input ID: ");

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

printf("Input Scores: ");

scanf("%d %d %d %d", &score[0][i], &score[1][i], &score[2][i], &score[3][i]);

}

//ID와 ID별 과목 점수 출력과 동시에 총합과 평균 계산 및 출력

printf("ID\tKOR\tENG\tMATH\tC++\tTOTAL\tAVG\n");

for (i = 0; i < 5; i++) {

printf("%d\t", ID[i]);

for (j = 0; j < 4; j++) {

printf("%d\t", score[j][i]);

sum += score[j][i];

}

average = sum / 4;

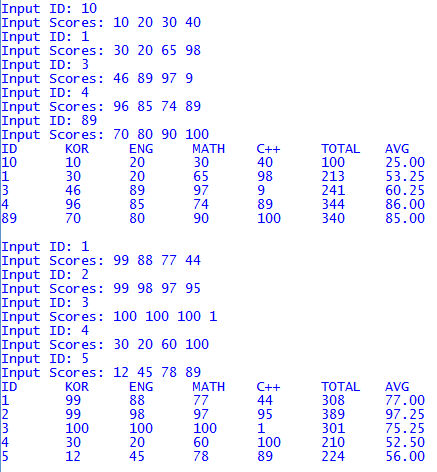
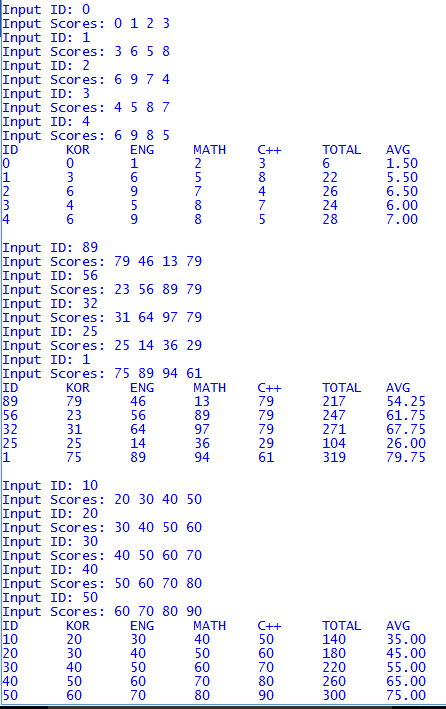
printf("%d\t%d\n", sum, average);

sum = 0;

}

}

**3.3. Result (snapshot)**



**4 홀수 마방진 만들기**

**4.1. Solution**

**마방진의 규칙과 예외 상황 등을 잘 정리하여 코드화 시키면 된다. 보통은 1차원 배열에서 현재 인덱스 + 1 그리고 밑으로 한 칸 내려가야 하기 때문에 r로 변의 길이만큼 +를 한다. 하지만 첫 번째로 생각해야할 예외가 그 자리에 숫자가 있을 때이다. 따라서 배열을 애초에 0으로 초기화 하고 넣고자 하는 자리가 0이 아닐 경우 그 위칸, 즉 인덱스 – 가로 변의 길이를 한다. 만약 현재 인덱스가 오른쪽 변의 끝에 있을 경우 인덱스 + 1만 하면 되고, 만약 배열의 끝부분이면 가장 첫 부분에 넣기만 하면 된다.**

**4.2. Source code**

void mabangjin()

{

//마방진 출력

int jin[10 \* 10] = { 0 }; //최대 넓이 100

int N, i;

int index;

scanf("%d", &N);

//입력 값이 짝수이면 N+1

if (N % 2 == 0)

N = N + 1;

//입력 값이 범위를 넘어가면 오류 문 출력

while (N > 10){

printf("N should be a number between 1 to 9\n");

printf("Try Again\n");

scanf("%d", &N);

}

//첫 value인 1의 index 계산

index = N\*N - (N / 2) - 1;

for (i = 1; i <= N\*N; i++) {

jin[index] = i;

//다음 index 값 계산

/\* 1.index+1의 값이 boundary일 시,

a) 배열의 첫번째 자리가 비어있을 때,

index는 0이 된다.

b) 배열의 첫번쨰 자리가 안비어있으면

index는 그림 상 그 윗칸(index - N)

c) 둘 다 해당사항 없으면 index + 1;

2. 다음 index값이 비어있는 공간이 아니면 index-N

3. default는 오른쪽 한칸 이동 후 밑으로 한 칸

\*/

if ((index + 1) % N == 0){

if ((index + 1 == N\*N)) {

if (jin[0] == 0)

index = 0;

else

index = index - N;

}

else

index = index + 1;

}

else if (jin[((index + 1) + N) % (N\*N)] != 0)

index = index - N;

else

index = ((index + 1) + N) % (N\*N);

}

//출력

for (i = 0; i < N\*N; i++) {

if (i%N == 0 && i != 0)

printf("\n");

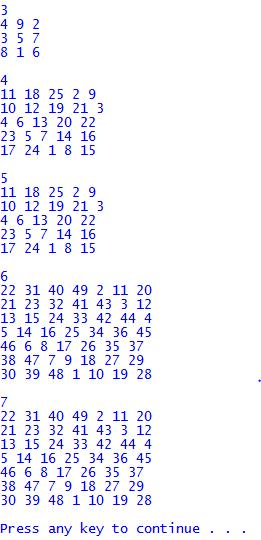
printf("%d ", jin[i]);

}

printf("\n");

}

**4.3. Result (snapshot)**



**5 같은 숫자 찾기**

**5.1. Solution**

**같은 수가 2번씩 나오기 때문에 배열의 크기/2만큼의 수 까지만 쓸 수 있다. 배열을 0으로 초기화 하고 1부터 수를 넣을 때 2개의 랜덤 인덱스를 계산하여 수를 삽입한다. 그리고 알맞게 출력한다.**

**5.2. Source code**

void SameNumberGame()

{

int set[10 \* 10] = { 0 }; //100넓이로 초기화(최대)

int width, height;

int area;

int index, i;

printf("Input width, height: ");

scanf("%d %d", &width, &height);

area = width \* height;

//넓이가 홀수 이면 오류 문 출력

if (area % 2 != 0)

{

while (1) {

printf("Area of the table should be even. Try Again\n");

printf("Input width, height: ");

scanf("%d %d", &width, &height);

area = width \* height;

if (area % 2 == 0)

break;

}

}

//1~전체 배열 원소 개수/2 까지 random index 2곳에 저장

for (i = 1; i <= area/2; i++) {

index = rand() % area;

if (set[index] != 0) {

//index값이 비어있지 않으면 빌 때까지 난수 생성

while (set[index] != 0)

index = rand() % area;

}

set[index] = i;

index = rand() % area;

if (set[index] != 0) {

//index값이 비어있지 않으면 빌 때까지 난수 생성

while (set[index] != 0)

index = rand() % area;

}

set[index] = i;

}

//출력

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

{

if (i % width == 0 && i != 0)

printf("\n");

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

}

printf("\n");

}

**5.3. Result (snapshot)**

