/\*

240318 순환

1번 - 순환 팩토리얼 계산(p.41)

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int factorial(int n)

{

if (n <= 1) return(1);

else return (n \* factorial(n-1));

}

/\*

240318 순환(p.54)

2번 - 피보나치 수열

\*/

#include<stdio.h>

#pragma warning(disable: 4996)

int fib\_iter(int n)

{

if (n == 0) return 0;

if (n == 1) return 1;

int pp = 0;

int p = 1;

int result = 0;

for (int i = 2; i <= n; i++) {

result = p + pp;

pp = p;

p = result;

}

return result;

}

int main()

{

int num, final\_result;

printf("Insert Integer :");

scanf("%d", &num);

final\_result = fib\_iter(num);

printf("\nthe result is %d\n", final\_result);

}

/\*

240318 순환(61p)

3번 - 하노이 탑

자,그러면 어떻게 n-1개의 원판을 A에서 B로, 또 B에서 C로 이동하는가?

(힌트) 우리의 원래 문제가 n개의 원판을 A에서 C로 옮기는 것임을 기억하라.

따라서 지금 작성하고 있는 함수의 매개변수를 n-1로 바꾸어 순환 호출하면 된다.

\*/

#include <stdio.h>

#pragma warning(disable: 4996)

void hanoi\_tower(int n, char from, char tmp, char to)

{

if (n == 1) printf("원판 1을 %c 에서 %c으로 옮긴다.\n", from, to);

else {

hanoi\_tower(n - 1, from, to, tmp);

printf("원판 %d을 %c에서 %c으로 옮긴다.\n", n, from, to);

hanoi\_tower(n - 1, tmp, from, to);

}

}

int main(void)

{

hanoi\_tower(4, 'A', 'B', 'C');

return 0;

}