자료구조 실습과제 02 _ 순환과 반복을 이용한 문제 해결

- 솔루션 및 프로젝트 명칭 : Proj_02_이름이니셜

- 제출방법 : 아래 문제를 해결하기 위한 프로그램을 구현한 후 컴파일 및 실행한 후, 오류가 없으면 메뉴에서 **솔루션 정리를 수행한 후 윈도우 탐색기에서 솔루션 폴더를 찾아 압축하여** E-class에 올림

  문제 해결 과정에서 현재 문제를 더 작은 데이터를 활용하여 동일한 방법으로 문제를 해결하는 경우에 순환 방법을 많이 활용한다. 이 방법은 반복적인 방법으로도 해결할 수 있다. 다만, 두 가지 방법을 적용하였을 경우 시간 복잡도에서 크게 차이나는 경우가 있으므로 주의해서 사용할 필요가 있다.

  이번 실습은 아래에 주어지는 문제를 **순환 방법과 반복적 방법으로 각각의 함수를 구현**하고, 데이터 크기가 n이라고 가정하였을 경우 동일한 n 값에 대하여 **두 방법의 시간 복잡도를 수행된 시간으로 비교하여 제시**한다. 또한 재귀적 문제 해결 방안에서 **함수의 호출 횟수를 출력**하여 데이터의 크기가 변할 때 어떤 관계로 계산량이 증가하는 확인하시오. 단, 재귀적으로 수행할 경우 함수 호출 깊이가 깊어질 경우 스택 제한으로 프로그램이 다운되는 경우가 발생할 수 있으니 가능한 범위 내에서 수행한다.

문제 1) 실수 x에 대하여 power(x, n)을 구하시오. 여기서 n은 0보다 큰 정수이다. 단, 수업시간에 검토한 일반적인 방법과 재귀적 방법을 활용하여 구현하시오.

문제 2) 이항 계수(binomial coefficient)를 계산하는 수식은 다음과 같다.

$$
{}_nC_k = \begin{cases} {}_{n-1}C_{k-1} + {}_{n-1}C_k & \text{if } 0 < k < n, \\ 1 & \text{if } k = 0 \text{ or } k = n \end{cases}
$$

문제 3) Ackerman 함수는 다음과 같이 정의된다. m이 4이상 되는 경우는 계산량이 매우 커지므로 중간에 프로그램이 비정상 종료될 수 있음.

$$
A(m, n) = \begin{cases} n+1 & \text{if } m = 0, \\ A(m-1, 1) & \text{if } m \neq 0 \ \&\& \ n = 0, \\ A(m-1, A(m, n-1)) & \text{otherwise} \end{cases}
$$

문제 4) 피보나치 수열은 다음과 같이 정의된다.

f(n) =	0	if n=0;
	1	if n=1;
	f(n-2) + f(n-1)	otherwise

문제 5) n >= 1 개의 서로 다른 정수가 이미 정렬되어 배열에 저장되어 있다. 여기서 검색하고자 하는 값 x를 입력받아 배열에서 검색하여 해당되는 위치를 돌려주는 프로그램을 작성하시오. 특별히 O(log n)에 검색할 수 있는 이진 검색 방법을 활용하여 구현하시오.

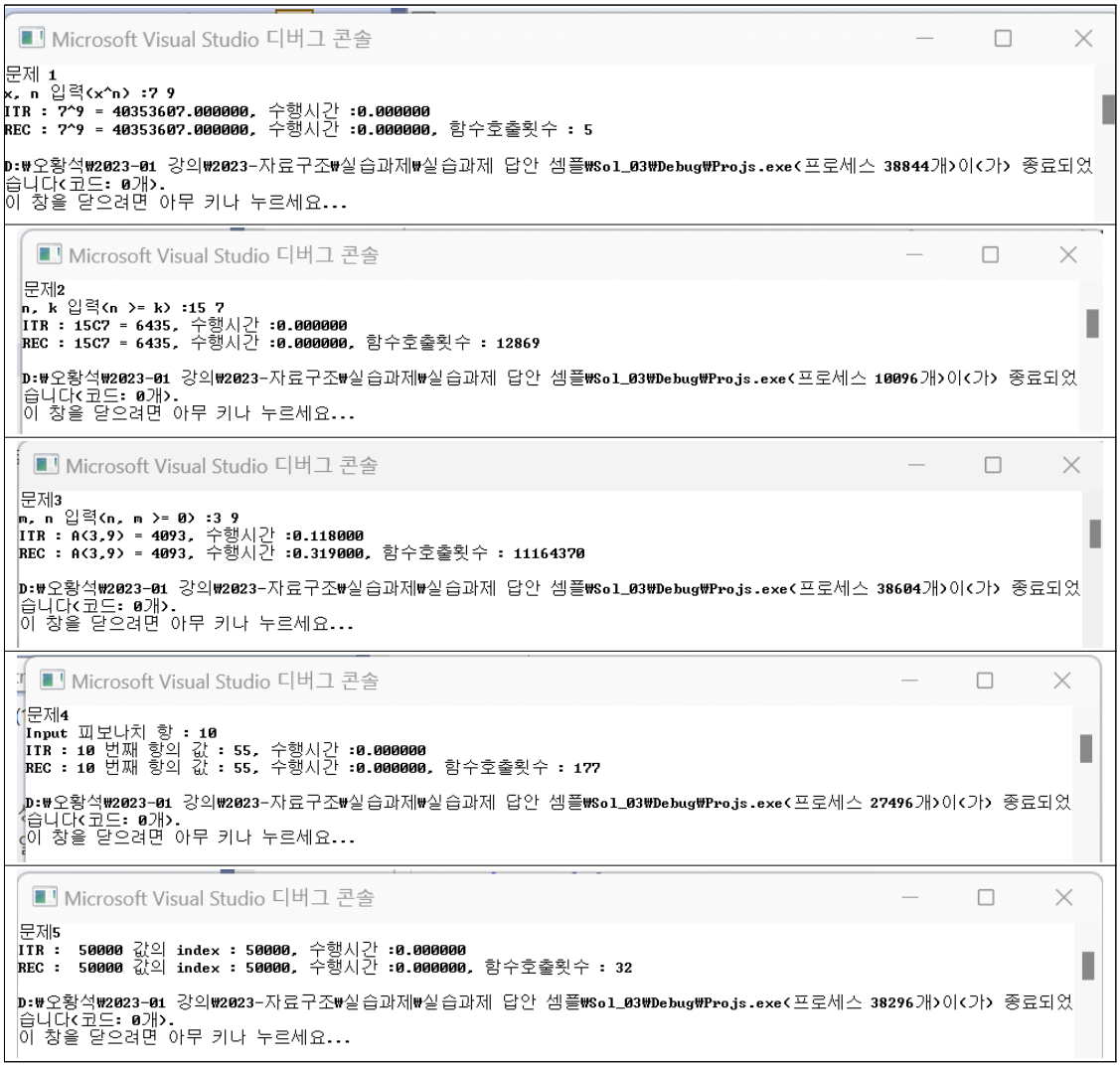
이진 검색 방법은 리스트의 중간 위치에 있는 값과 입력값 x를 비교하여 일치하면 그 위치를 돌려주고, 찾는 값이 중간 위치의 값보다 작은 경우 중간 위치의 앞쪽 부분을 대상으로 같은 방법으로 검색하고, 그렇지 않을 경우 중간 위치의 뒷 부분을 대상으로 같은 방법으로 검색한다.

n 값을 100, 1,000, 10,000, 100,000, 1,000,000을 대상으로 배열의 각 위치에 인덱스 값을 저장하고 검색하는 데이터로 활용하시오. 예를 들어
#define SIZE 1000
...
for(idx=0; idx<SIZE; idx++) data[i] = idx;
로 초기화하고, x 값을 입력받아 해당 값을 검색하시오.

프로그램 개발 구조 예

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <time.h>

int rec_count = 0; //재귀적 함수 호출 횟수를 저장하기 위한 함수

#define PROB 1 // 각각의 문제를 구현하고 해당 문제 번호를 변경하여 테스트

#if PROB == 1
/* Compute power(x, n) for int x, n */
// 1번 문제에 대한 반복, 재귀적 방법으로 구현하기 위한 각각의 함수 구현
// main 함수에서 데이터를 입력받아 호출하는 코드와 결과 출력 구현

#elif PROB == 2
/* Binomial Coef */
// 2번 문제에 대한 반복, 재귀적 방법으로 구현하기 위한 각각의 함수 구현
// main 함수에서 데이터를 입력받아 호출하는 코드와 결과 출력 구현

#elif PROB == 3
/* Ackerman function */


#elif PROB == 4
/* 문제 3) 피보나치 수열 */


#elif PROB == 5
/* 문제 4) Binary Search */


#endif
```

실행 예


Microsoft Visual Studio 디버그 콘솔
문제 1
x, n 입력<x^n> :7 9
ITR : 7^9 = 40353607.000000, 수행시간 :0.000000
REC : 7^9 = 40353607.000000, 수행시간 :0.000000, 함수호출횟수 : 5
D:₩오황석₩2023-01 강의₩2023-자료구조₩실습과제₩실습과제 답안 셈플₩Sol_03₩Debug₩Projs.exe<프로세스 38844개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...
Microsoft Visual Studio 디버그 콘솔
문제2
n, k 입력<n >= k> :15 7
ITR : 15C7 = 6435, 수행시간 :0.000000
REC : 15C7 = 6435, 수행시간 :0.000000, 함수호출횟수 : 12869
D:₩오황석₩2023-01 강의₩2023-자료구조₩실습과제₩실습과제 답안 셈플₩Sol_03₩Debug₩Projs.exe<프로세스 10096개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...
Microsoft Visual Studio 디버그 콘솔
문제3
m, n 입력<n, m >= 0> :3 9
ITR : A(3,9) = 4093, 수행시간 :0.118000
REC : A(3,9) = 4093, 수행시간 :0.319000, 함수호출횟수 : 11164370
D:₩오황석₩2023-01 강의₩2023-자료구조₩실습과제₩실습과제 답안 셈플₩Sol_03₩Debug₩Projs.exe<프로세스 38604개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...
Microsoft Visual Studio 디버그 콘솔
문제4
Input 피보나치 항 : 10
ITR : 10 번째 항의 값 : 55, 수행시간 :0.000000
REC : 10 번째 항의 값 : 55, 수행시간 :0.000000, 함수호출횟수 : 177
D:₩오황석₩2023-01 강의₩2023-자료구조₩실습과제₩실습과제 답안 셈플₩Sol_03₩Debug₩Projs.exe<프로세스 27496개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...
Microsoft Visual Studio 디버그 콘솔
문제5
ITR : 50000 값의 index : 50000, 수행시간 :0.000000
REC : 50000 값의 index : 50000, 수행시간 :0.000000, 함수호출횟수 : 32
D:₩오황석₩2023-01 강의₩2023-자료구조₩실습과제₩실습과제 답안 셈플₩Sol_03₩Debug₩Projs.exe<프로세스 38296개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...