

과목명	자료구조 및 알고리즘	분반	x	담당교수	김화성 교수님
학과	전자통신공학과	학번	2016707079	이름	하상천
H/W 1 : Fibonacci					

1, 과제설명 (과제에 대한 설명 및 목표)

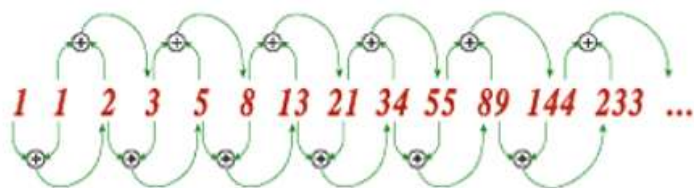
(1) Fibonacci Series 계산 프로그램을 Recursion과 Iteration을 이용하여 각각 구현하라.
(C 언어로 작성)

(1)번에서 구현된 Recursion 버전과 Iteration 버전을 각각 이용하여 Fib(n)의 값을 프린트할 수 있도록 부가적인 코드를 작성하고, 실행결과를 화면 캡처하여 제출하세요.

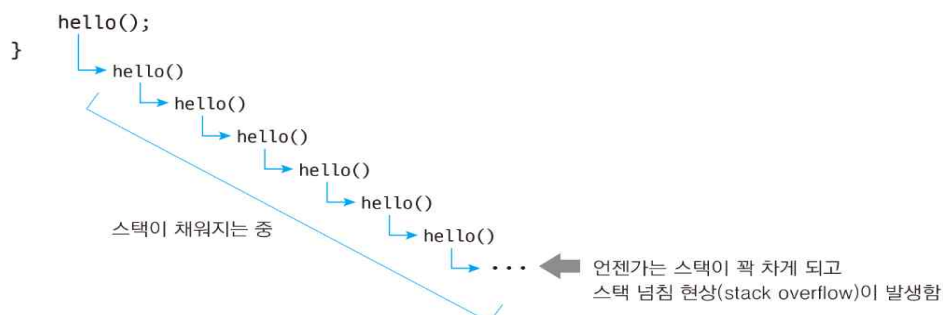
2, 이론 (과제와 관련한 일반적인 내용)

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

3. 알고리즘 및 자료구조 설계 내용 기술



```
void hello()
{
    printf("Hello, world!\n");
}
```



피보나치 수열 계산을 recursion 버전과 iteration 버전으로 구현해보고, 시간 복잡도를 통해 예상 실행 시간을 구해본다.

4. 소스코드 설명 (직접 작성한 소스코드중에 핵심 부분을 발췌하여 설명)

```
#include<time.h>
```

시간 측정을 위한 라이브러리이다.

```
printf(", %4d", fib(looper));
```

결과값이 4칸이 안될 때에도 임의적으로 4칸을 잡아주어서 숫자간의 간격을 유지한다.

```
start = clock();  
end = clock(); //시간 측정 끝  
result = (float)(end - start) / CLOCKS_PER_SEC;  
printf("소요된시간: %f", result);
```

프로그램이 시작할 때 시간을 측정하고, 끝날 때 측정하여 프로그램의 실행시간을 측정하였다.

```
int fib(int num) {  
    if (num == 0 || num == 1) {  
        return num;  
    }  
    return(fib(num - 1) + fib(num - 2));  
}
```

recursion을 이용하여 피보나치 수열을 계산하였다.

```
int fib2(int num) {  
    int a = 0;  
    int b = 1;  
    int temp;  
    for (int i = 0; i < num - 1; i++) {  
        temp = a+b;  
        a = b;  
        b = temp;  
    }  
    return b;  
}
```

iteration을 이용하여 피보나치 수열을 계산하였다.

5. 실행결과 및 설명 (실행 결과를 캡처하여 첨부한 후 설명)

```
Microsoft Visual Studio 디버그 콘솔
Input numbers : 30
Print a Fibonacci series (Recursion).

0, 1, 1, 2, 3
5, 8, 13, 21, 34
55, 89, 144, 233, 377
610, 987, 1597, 2584, 4181
6765, 10946, 17711, 28657, 46368
75025, 121393, 196418, 317811, 514229
소요된 시간 : 6.414000
C:\Users\seomk\source\repos\Project73\Debug\Project73.exe(13444 프로세스)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[콘솔 설정]을 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

```
Microsoft Visual Studio 디버그 콘솔
Input numbers : 30
Print a Fibonacci series (Iteration).

0, 1, 1, 2, 3
5, 8, 13, 21, 34
55, 89, 144, 233, 377
610, 987, 1597, 2584, 4181
6765, 10946, 17711, 28657, 46368
75025, 121393, 196418, 317811, 514229
소요된 시간 : 3.113000
C:\Users\seomk\source\repos\Project72\Debug\Project72.exe(11624 프로세스)이(가) ,
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이
콘솔 설정]을 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

(그림을 문서에 포함, 글자처럼 취급 옵션, 잉크 절약과 잘 보이게 하기 위해 그림 반전)

Recursion 버전과 Iteration 버전을 각각 이용하여 Fib(n)의 값을 출력하는 실행결과 화면이다.

6. 고찰 (과제를 진행하면서 배운점 이나, 시행 착오 내용, 기타 느낀점)

이번 과제는 피보나치 수열 계산 프로그램을 recursion과 iteration을 이용하여 각각 구현하는 것이었다. recursion 버전은 예전에 많이 봐서 쉽게 구현하였지만, iteration 버전은 처음에 생각이 잘 나지 않았다. 그래서 노트에 숫자를 하나씩 적어가면서 생각해봤더니 조금 더 쉽게 구현할 수 있었던 것 같다. recursion 버전과 iteration 버전을 빅오 표기법의 시간 복잡도로 표현해보니 recursion 버전은 $O(2^n)$ 이고, iteration 버전은 $O(n)$ 이었다. 실제로 recursion 버전이 시간이 더 오래걸리는지 확인하기

위해 time.h를 include 한 후 프로그램을 시작할 때 시간을 측정하고 프로그램이 끝날 때 시간을 측정하여 두 시간의 차이로 실행시간을 알아보았다. 위의 실행 결과창에서 확인할 수 있듯이 recursion 버전이 iteration 버전보다 약 2배정도 오래 걸렸음을 알 수 있었다. 예전에는 프로그램을 구현하고 실행이 되면 좋아했었는데, 이제부터는 좋은 개발자가 되기 위해서 실행시간 뿐만 아니라 메모리 낭비 같은 문제도 생각하면서 프로그램을 구현해야 될 것 같다.

7. 전체 소스코드 (글자크기 9에 줄간격을 120%로 유지하고 한 줄이 너무 길지 않게 작성)

```
<Recursion>

#include<stdio.h>
#include<time.h> //시간 측정을 위한 라이브러리
int fib(int num);
int main() {
    clock_t start, end;
    float result;
    start = clock(); //시간 측정 시작
    int seriesSize;
    printf("Input numbers : ");
    scanf("%d", &seriesSize); //숫자를 입력받는다.

    printf("Print a Fibonacci series (Recursion). \n");

    for (int loop = 0; loop < seriesSize; loop++) {
        if (loop % 5) // 숫자5개씩 끊어서 줄을 바꾸었다.
            printf(", %4d", fib(loop)); // %4d를 이용하여 숫자끼리 띄어주었다.
        else
            printf("\n%4d", fib(loop));
    }
    printf("\n");
    end = clock(); //시간 측정 끝
    result = (float)(end - start) / CLOCKS_PER_SEC;
    printf("소요된시간: %f", result); // 소요된 시간출력
    return 0;

}

int fib(int num) { //재귀를 이용한 피보나치 수열 계산
    if (num == 0 || num == 1) {
        return num;
    }
    return(fib(num - 1) + fib(num - 2));
}
```

<Iteration>

```
#include<stdio.h>
#include<time.h> //시간 측정을 위한 라이브러리

int fib(int num);
int fib2(int num);
int main() {
    clock_t start, end;
    float result;
    start = clock(); //시간 측정 시작
    int seriesSize;
    printf("Input numbers : ");
    scanf("%d", &seriesSize);//숫자를 입력받는다.

    printf("Print a Fibonacci series (Iteration). \n");

    for (int loop = 0; loop < seriesSize; loop++) {
        if (loop % 5) // 숫자5개씩 끊어서 줄을 바꾸었다.
            printf(", %4d", fib(loop)); // %4d를 이용하여 숫자끼리 띄어주었다.
        else
            printf("\n%4d", fib(loop));
    }
    printf("\n");
    end = clock(); //시간 측정 끝
    result = (float)(end - start) / CLOCKS_PER_SEC
    printf("소요된시간: %f", result);
    return 0;
}

int fib(int num) {
    int result;
    if (num == 0 || num == 1) {
        return num;
    }
    else {
        result = fib2(num); //피보나치 수열 계산함수 호출
    }
    return result;
}

int fib2(int num) { //Iteration을 이용한 피보나치 수열 계산
    int a = 0;
    int b = 1;
    int temp;
```

```
for (int i = 0; i < num - 1; i++) {  
temp = a+b;  
a = b;  
b = temp;  
}  
return b;  
}
```

(글자크기는 10으로 유지하고 줄간격도 160%를 유지할 것)