5645640 제갈건 0314

1. 실행코드

#include<stdio.h>

#include<time.h>

int factorial\_recur(int n) { // Recursion함수

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

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

int k = 0;

for (int i = 1; i <= 1000; i++) { // 소요시간 측정을 위한 더미코드

k++;

}

}

void factorial\_iter(int n) {

clock\_t start = clock(); // 함수의 시작 시간 저장

int result = 1, i, k = 0;

for (i = 1; i <= n; i++) { // 펙토리얼 실행

result = result \* i;

}

for (i = 1; i <= 1000; i++) { // 소요시간 측정을 위한 더미코드

k++;

}

clock\_t finish = clock(); // 함수의 종료시간 저장

double duration = (double)(finish - start) / CLOCKS\_PER\_SEC; // Iteration함수의 소요시간 계산

printf("factorial\_iter : %d , time : % lf", result, duration);

}

int main() {

clock\_t start = clock(); // Recursion함수의 시작 시간 저장

int result = factorial\_recur(3220); // 함수 실행

clock\_t finish = clock(); // Recursion함수의 종료 시간 저장

double duration = (double)(finish - start) / CLOCKS\_PER\_SEC;

// Recursion함수의 소요시간 계산

printf("factorial\_recur : %d ,time : % f\n", result, duration);

// Recursion함수의 결과값과 소요시간 출력

// factorial\_iter(3220); // Iteration함수

return 0;

}

1. 실행화면

텍스트이(가) 표시된 사진

자동 생성된 설명

(recursion방식과 iteration방식의 비교)

텍스트이(가) 표시된 사진

자동 생성된 설명

(3220에서 stack overflow 발생)

텍스트이(가) 표시된 사진

자동 생성된 설명

(recursion방식에서 stack overflow가 발생한 숫자가 iteration방식에서는 정상적으로 작동함)

1. 고찰

처음 디버깅을 하였을 때는 결과값이 양수에서 음수로 바뀌는 것(자료형이 int라서 표현하지 못하는 것)을 stack overflow로 착각하였습니다. 이후 교수님의 조언을 받고 입력 값을 키워 stack overflow가 정상적으로 일어나는 것을 확인하였습니다.