**<함수 설명 및 실행 결과>**

**1. 함수 설명 – 링크**

[**static** **void** NewFile(String path)](#NewFile)

[**static** **void** UpdateFile(String path, String Text)](#UpdateFile)

[**static** String reverse(String str)](#reverse)

[**static** **int** getInt(String a, **int** i)](#getInt)

[**static** String Plus(String a, String b)](#Plus)

[**static** String leftShift(String a)](#leftShift)

[**static** String rightShift(String aa)](#rightShift)

[**static** String multiply(String a, String b)](#multiply)

[**static** String Minus(String aa, String bb)](#Minus)

[**static** String divideMul(Vector<String> list, **int** n)](#divideMul)

[**static** String Factorial(**int** n)](#Factorial)

[**static** String karatsuba(String a, String b)](#karatsuba)

[**public** **static** **void** main(String[] args)](#main)

**static** **void** NewFile(String path)

path에 새로운 파일을 생성해주는 함수입니다.

입력 값은 파일의 경로입니다.

경로상에 있는 폴더들은 이미 만들어져 있어야 합니다.

**static** **void** UpdateFile(String path, String Text)

입력 : 파일의 경로, 문자열

path에 있는 파일에 Text문자열을 새로 write해주는 함수입니다.

path에 파일이 존재하지 않는 경우, 새로 생성해줍니다.

**static** String reverse(String str)

입력 : 문자열 / 출력 : 문자열의 역순

입력한 문자열 str의 역순 문자열을 반환해주는 함수입니다.

문자열의 연산 과정에서 사용합니다.

**static** **int** getInt(String a, **int** i)

문자열 a의 i번째 인덱스 값을 숫자 값으로 변환해서 돌려주는 함수입니다.

문자열 연산을 위해 만들었습니다.

**static** String Plus(String a, String b)

두 스트링 a, b의 합을 문자열 형태로 반환하는 함수입니다.

a, b의 각 자리수에 맞춰서 덧셈결과 및 캐리를 처리하고, 최종 결과값을 반환하는 로직으로 구현했습니다.

이 함수는 reverse된 값을 받고, reverse된 값을 반환하도록 구현했기 때문에, a+b의 결과를 얻기 위해서는 reverse(Plus(reverse(a), reverse(b)))의 형태로 사용해야 합니다.

**static** String leftShift(String a)

문자열 a를 왼쪽으로 시프트연산한 결과를 문자열 형태로 반환합니다.

문자열의 각 글자마다 시프트 연산과 캐리처리를 진행했습니다.

a<<1 의 결과를 얻고자 할 경우, reverse(leftShift(reverse(a)))의 형태로 사용합니다.

**static** String rightShift(String aa)

문자열 a를 오른쪽으로 시프트연산한 결과를 문자열 형태로 반환합니다.

문자열의 각 글자마다 시프트 연산과 캐리처리를 진행했습니다.

a>>1 의 결과를 얻고자 할 경우, reverse(rightShift(reverse(a)))의 형태로 사용합니다.

**static** String multiply(String a, String b)

a\*b의 결과값을 문자열로 반환하는 함수입니다.

a la russe 알고리즘을 이용했으며, 자리수가 적은 곱셈연산을 수행할 때 이용됩니다.

사용 시, a\*b를 구하기 위해 reverse(multiply(reverse(a), reverse(b))) 형태로 사용됩니다.

**static** String Minus(String aa, String bb)

aa-bb의 결과를 문자열 형태로 반환합니다.

이 과제에서는 더 큰 값을 빼는 경우가 없으므로, 항상 aa가 bb보다 큰 값이라고 가정했습니다.

이 함수는 정상 값을 받아서, 함수 내에서 reverse를 취한 후에 각 자리에 따른 뺄셈을 진행합니다. 따라서 사용시에 Minus(aa, bb) 형태로 사용됩니다.

**static** String divideMul(Vector<String> list, **int** n)

list에 있는 문자열 값들을 모두 곱한 결과를 문자열 형태로 반환하는 함수입니다.

N은 현재 list 내에 있는 문자열들의 최소 자릿수를 의미하며, 이를 기준으로 divide and conquer를 이용한 곱셈을 수행합니다. 이 함수는 factorial함수를 구현할 때, 비슷한 자리수끼리 연산을 하기 위해 제작되었습니다.

먼저, String들을 저장하는 벡터 over을 선언합니다. 이후 list내의 값들을 계속해서 곱합니다. 만약 중간결과의 자리수가 2\*n 이상이거나 마지막 요소인 경우, over에 그 값을 넣고 divideMul(over, 2\*n)을 반환합니다.

**static** String Factorial(**int** n)

n!의 결과값을 문자열 형태로 반환하는 함수로, 주어진 과제의 최종 목표인 50000!을 계산해주는 함수입니다.

먼저 처음에는 50000부터 2까지 곱해가면서 결과값의 자리수가 10이 넘어가면, Vector<String> list에 삽입하고 현재결과를 1로 초기화합니다.

이 과정이 끝나면, 현재의 결과값을 list에 포함시키기 위해 list의 0번째 요소에 곱합니다.

그러면 list에는 10자리 이상의 문자열들이 저장되어 있을 것입니다. 이 list로 divideMul(list, 10)을 반환함으로써 최종 결과값이 반환됩니다.

**static** String karatsuba(String a, String b)

a\*b의 결과값을 문자열 형태로 반환해주는 함수로, 카라추바 알고리즘을 이용해서 구현했습니다. 자리수가 큰 문자열끼리 곱셈을 하기 위해 이를 구현했으며, 자리수가 낮은 문자열에 대해서는 multiply 함수를 수행합니다.

**public** **static** **void** main(String[] args)

메인 함수에서는 파일 저장경로인 String path = "C:\\homework\\answer.dat"; 를 선언해주고, 이후에 UpdateFile(path, Factorial(50000)); 의 수행시간을 출력해줍니다. 결과값은 c\:homework 폴더에 answer.dat로 저장됩니다.

**2. 실행 결과**

-콘솔 실행결과 화면 : 중간과정 및 수행시간을 출력한 결과입니다. “현재 : 00”은 factorial함수에서 list에 문자열을 삽입하면서 출력한 것으로, 초기에 50000부터 시작되어서2까지 출력된 형태입니다. 이후 divideMul함수에서 분할정복을 시작하여, 각 divideMul함수마다 n값과 list의 size를 출력한 모습입니다. 그리고 최종적으로 프로그램 수행시간을 출력했습니다. 총 2388초로, 계산하는 데에 약 40분정도가 소요되었습니다.

