#include <stdio.h>

#include <stdlib.h>

#include <windows.h> //QueryPerformanceFrequency()

float\* SlowprefixAverages(int n, int \*X, float \*A); //느린버전의 함수

float\* FastprefixAverages(int n, int \*X, float \*A); //빠른버전의 함수

int main() {

int n[3] = { 100000, 200000, 300000 }; //측정해야하는 n의 세 수를 배열로 정의

int i; //for()

int \*X; //

float \*A; //

LARGE\_INTEGER Frequency; //QueryPerformanceFrequency()를 사용하기위한 빈도 정의

LARGE\_INTEGER startTime, endTime; //QueryPerformanceFrequency()를 사용하기위한 시작시간과 종료시간 정의

for (i = 0; i < 3; i++) { //n = 십만, 이십만, 삼십만으로 세버납ㄴ복

A = (float \*)malloc(sizeof(float) \* n[i]); //포인터 A의 동적할당

X = (int \*)malloc(sizeof(int) \* n[i]); //포인터 X의 동적할당

srand((unsigned int)time(NULL)); //난수발생

QueryPerformanceFrequency(&Frequency); //ms단위까지 알려주는 시간측정함수

QueryPerformanceCounter(&startTime); //느린버전 함수 실행전의 시간측정

SlowprefixAverages(n[i], X, A); //느린버전 함수 실행

QueryPerformanceCounter(&endTime); //함수 실행 후의 시간 측정

printf("n의 값이 %d인 경우 slow 실행시간 : %f초\n", n[i], (double)(endTime.QuadPart - startTime.QuadPart) / Frequency.QuadPart);

QueryPerformanceCounter(&startTime); //빠른버전 함수 실행전의 시간측정

FastprefixAverages(n[i], X, A);; //빠른버전 함수 실행

QueryPerformanceCounter(&endTime); //함수 실행 후의 시간 측정

printf("n의 값이 %d인 경우 fast 실행시간 : %lf초\n", n[i], (double)(endTime.QuadPart - startTime.QuadPart) / Frequency.QuadPart);

printf("\n\n");

}

return 0;

}

float\* SlowprefixAverages(int n, int \*X, float \*A) { //느린 버전

int i, j;

float sum = 0.0;

for (i = 0; i < n; i++) {

sum++;

for (j = 0; j < i; j++)

sum += X[j];

A[i] = sum / (float)(i + 1);

}

return A;

} //slow

float\* FastprefixAverages(int n, int \*X, float \*A) { //빠른 버전

int i;

float sum = 0.0;

for (i = 0; i < n; i++) {

sum += X[i];

A[i] = sum / (float)(i + 1);

}

return A;

} //fast