#include <iostream>

const int SIZE = 12 \* 1024 \* 1024 / 4;

const int OFFSET = 24 \* 1024 \* 1024 / 4;

extern \_\_inline\_\_ uint64\_t rdtsc\_x(){

uint64\_t x;

\_\_asm\_\_ volatile ("rdtsc\n\tshl $32, %%rdx\n\tor %%rdx, %%rax" : "=a" (x) : : "rdx");

return x;

}

int main(){

int\* arr = new int[OFFSET \* 32];

int newSize = 0;

uint64\_t start = 0;

uint64\_t end = 0;

uint64\_t max = 0;

uint64\_t diff = 0;

for (int N = 1; N <= 32; N++){

newSize = SIZE / N;

for (int i = 0; i < newSize; i++){

for (int k = 0; k < N - 1; k++){

arr[i + OFFSET \* k] = i + OFFSET \* (k + 1);

}

arr[i + OFFSET \* (N - 1)] = i + 1;

}

max = UINT64\_MAX;

for (int ii = 0; ii < 25; ii++){

start = rdtsc\_x();

for (int k = 0, jj = 0; jj < newSize \* N; jj++){

k = arr[k];

}

end = rdtsc\_x();

diff = end - start;

if (diff < max){

max = diff;

}

}

std::cout << N << " fragments ----- " << diff/(newSize \* N) << " tacts" << std::endl;

}

delete[] arr;

return 0;

}