НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ ТА СПЕЦІАЛІЗОВАНИХ КОМП’ЮТЕРНИХ СИСТЕМ

**Лабораторна робота №5  
з дисципліни «Системне програмне забезпечення»**

**Варіант 9**

Виконав  
студент 4-го курсу  
групи КВ-41  
Курач Віктор

Київ – 2017

**Постановка задачі**

1. Проаналізувати програму, виміряти час її виконання, визначити, де є простір для поліпшення.
2. Виходячи з описаних у теорії механізмів організації пам'яті внести необхідні зміни в структуру коду.
3. Виміряти новий час виконання, за наявності очевидних поліпшень, переходити до п.4, якщо немає - до п.1
4. Зафіксувати нові показники і пояснити, за рахунок чого було досягнуто прискорення.

**Завдання для даного варіанту**

static void Main(string[] args) {

int[,,] a = new int[100,100];

int res = 0;

for (int i = 1; i < 100; i++)

{

for (int j = 1; j < 100; j++)

{

a[j,i] = a[j-1,i-1]+1;

}

}

for (int i = 0; i < 100; i++)

{

for (int j = 0; j < 100; j++)

{

a[j,i]++;

}

}

}

**Неоптимізована програма**

#include <time.h>

#include <stdio.h>

#include <malloc.h>

#define N 10000

void main() {

int i, j;

time\_t start, end;

int \*\*a = (int \*\*)malloc(N\*sizeof(int \*));

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

a[i] = (int \*)malloc(N\*sizeof(int));

}

start = time(0);

for (i = 1; i < N; i++) {

for (j = 1; j < N; j++) {

a[j][i] = a[j-1][i-1] + 1;

}

}

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

for (j = 0; j < N; j++) {

a[j][i]++;

}

}

end = time(0);

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

free(a[i]);

}

free(a);

printf("%.20f\n", difftime(end, start));

}

**Оптимізована програма**

#include <time.h>

#include <stdio.h>

#include <malloc.h>

#define N 10000

void main() {

int i, j;

time\_t start, end;

int \*\*a = (int \*\*)malloc(N\*sizeof(int \*));

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

a[i] = (int \*)malloc(N\*sizeof(int));

}

start = time(0);

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

for (j = 0; j < N; j++) {

if (i > 0 && j > 0) a[i][j] = a[i-1][j-1] + 1;

a[i][j]++;

}

}

end = time(0);

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

free(a[i]);

}

free(a);

printf("%.20f\n", difftime(end, start));

}

**Висновки**

Час виконання першої програми – близько **6 с**, другої, оптимізованої – близько **1 с**. Оптимізація полягає в наступному:

1. В оптимізованій програмі замість обходу матриці по стовпцях виконується обхід по рядках. Елементи масиву розміщуються в пам’яті «по рядках», тому менше часу витрачається на обчислення адрес (зміщень), до яких звертається програма.
2. В оптимізованій програмі реалізовано один цикл замість двох. Можна припустити, що в даному випадку це зменшує час виконання програми як мінімум вдвічі.