

Rozważmy następującą procedurę uruchamiającą eliminację Gaussa bez pivotingu

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 1000
#include <sys/time.h>
#include <time.h>

static double gtod_ref_time_sec = 0.0;

/* Adapted from the bl2_clock() routine in the
BLIS library from Robert van de Geijn*/

double dclock()
{
    double the_time, norm_sec;
    struct timeval tv;
    gettimeofday( &tv, NULL );
    if ( gtod_ref_time_sec == 0.0 )
        gtod_ref_time_sec = ( double ) tv.tv_sec;
    norm_sec =
        ( double ) tv.tv_sec -gtod_ref_time_sec;
    the_time = norm_sec + tv.tv_usec * 1.0e-6;
    return the_time;
}
```

```

int GE(double A[][SIZE])
{
    int i,j,k;
    for (k = 0; k < SIZE; k++) {
        for (i = k+1; i < SIZE; i++) {
            for (j = k+1; j < SIZE; j++) {
A[i][j]=A[i][j]-A[k][j]*(A[i][k]/A[k][k]);
            }
        }
    }
    return 0;
}

```

```

int main( int argc, const char* argv[] )
{
    int i,j,iret;
    double matrix[SIZE][SIZE];
    double dtime;
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            matrix[i][j]=i+j;
        }
    }
}

```

```
    dtime = dclock();  
    iret=GE(matrix);  
    dtime = dclock()-dtime;  
    printf( "Time: %le \n", dtime);  
    fflush( stdout );  
  
    return iret;  
}
```

1. Proszę skompilować procedurę bez optymalizacji

```
paszynsk@atari:~/optimize/GE_dtime$ gcc ge1.c
```

```
paszynsk@atari:~/optimize/GE_dtime$ ./a.out
```

```
Time: 3.243409e+00
```

```
paszynsk@atari:~/optimize/GE_dtime$
```

2. Proszę wykonać kopię procedury

```
paszynsk@atari:~/optimize/GE_dtime$ cp ge1.c ge2.c
```

oraz dokonać w ge2.c pierwszej próby optymalizacji procedury GE poprzez umieszczenie zmiennych kierujących pętlami w rejestrach

```
register unsigned int i, j, k;
```

Proszę skompilować procedurę z pierwszą optymalizacją

Jaki jest czas działania?

3. Proszę wykonać kopię procedury

```
paszynsk@atari:~/optimize/GE_dtime$cp ge2.c ge3.c
```

oraz dokonać w ge3.c drugiej próby optymalizacji procedury GE poprzez wykonanie lokalnej kopii oraz umieszczenie rozmiaru pętli w rejestrach

```
register unsigned int local_size;
```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

4. Proszę wykonać kopię procedury

```
paszynsk@atari:~/optimize/GE_dtime$cp ge3.c ge4.c
```

oraz dokonać w ge4.c trzeciej próby optymalizacji procedury GE poprzez zamianę pętli względem i oraz j z krokiem +1

```
for (k = 0; k < local_size; k++) {
    for (i = k+1 ; i < local_size; i++) {
        for (j = k+1 ; j < local_size ; j++ ) {
            A[i][j]=A[i][j]-A[k][j]*(A[i][k]/A[k][k]);
        }
    }
}
```

na pętle względem i oraz j z krokiem +4

```

for (k = 0; k < local_size; k++) {
    for (i = k+1; i < local_size; ) {
        for (j = k+1; j < local_size; ) {
            if(j<local_size-4 && i<local_size-4)
                {
// Tu proszę uzupełnić

                    i=i+4; j=j+4;
                }
            else
                {
A[i][j]=A[i][j]-A[k][j]*(A[i][k]/A[k][k]);
                    i++; j++;
                }
        }
    }
}

```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

5. Proszę wykonać kopię procedury

```
paszynsk@atari:~/optimize/GE_dtime$cp ge4.c ge5.c
```

oraz dokonać w ge5.c czwartej próby optymalizacji procedury GE poprzez umieszczenie wspólnie używanych zmiennych wewnątrz pętli w rejestrach

```
register double XXX;
```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

6. Proszę przerobić program tak żeby wykonywał testy w pętli dla zwiększających się rozmiarów macierzy od 100 do 5000 z krokiem co 100 i narysować wykresy dla poszczególnych etapów optymalizacji.