Rozważmy następującą procedurę uruchamiającą mnożenie macierzy

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 500
#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 mm(double first[][SIZE], double
second[][SIZE], double multiply[][SIZE])
{
  int i,j,k;
  double sum = 0;
  for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++) {
      for (k = 0; k < SIZE; k++) {
         sum = sum + first[i][k]*second[k][j];
          }
          multiply[i][j] = sum;
          sum = 0;
    }
  }
  return 0;
}

int main( int argc, const char* argv[] )
{
  int i,j,iret;
  double first[SIZE][SIZE];
  double second[SIZE][SIZE];
  double multiply[SIZE][SIZE];
```

```
  for (i = 0; i < SIZE; i++) { //rows in first
    for (j = 0; j < SIZE; j++) { //columns in
first
      first[i][j]=i+j;
      second[i][j]=i-j;
    }
  }
  dtime = dclock();
  iret=mm(first,second,multiply);
  dtime = dclock()-dtime;
  printf( "Time: %le \n", dtime);
  fflush( stdout );
  return iret;
}
```

1. Proszę skompilować procedurę bez optymalizacji

paszynsk@atari:~/optimize/MM_dtime$ gcc mm1.c

paszynsk@atari:~/optimize/MM_dtime$ ./a.out

Time: 8.294200e-01

paszynsk@atari:~/optimize/MM_dtime$

2. Proszę wykonać kopię procedury

paszynsk@atari:~/optimize/MM_dtime$cp mm1.c mm2.c

oraz dokonać w mm2.c pierwszej próby optymalizacji procedury MM 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/MM_dtime$cp mm2.c mm3.c

oraz dokonać w mm3.c drugiej próby optymalizacji procedury MM 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/MM_dtime$cp mm3.c mm4.c

oraz dokonać w mm4.c trzeciej próby optymalizacji procedury MM poprzez zamianę pętli z krokiem +1

```
  for (i = 0; i < local_size; i++) {
    for (j = 0; j < local_size; j++) {
       for (k = 0; k < local_size; k++) {
          sum = sum + first[i][k]*second[k][j];
          }
```

```
            multiply[i][j] = sum;
            sum = 0;
      }
   }
```

na pętle kończące się zerowaniem licznika

```
   for (i = SIZE; i-- ; )
```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

5. Proszę wykonać kopię procedury

paszynsk@atari:~/optimize/MM_dtime$cp mm3.c mm5.c

oraz dokonać w mm5.c kolejnej próby optymalizacji procedury MM poprzez zamianę pętli względem k z krokiem -1

```
   for (i = 0; i < local_size; i++) {
     for (j = 0; j < local_size; j++) {
        for (k = 0; k < local_size; k++) {
         sum = sum + first[i][k]*second[k][j];
           }
         multiply[i][j] = sum;
          sum = 0;
     }
   }
```

Na pętle względem k z krokiem -8

```
  for (i = 0; i < local_size; i++) {
    for (j = 0; j < local_size; j++) {
       for (k = 0; k < local_size; ) {
        if(k<SIZE-8) {
//TU PROSZĘ UZUPEŁNIĆ
          k=k+8;
        }
        else {
          sum = sum + first[i][k]*second[k][j];
          k++;
        }
        multiply[i][j] = sum;
         sum = 0;
      }
    }
  }
```

Proszę skompilować procedurę z pierwszą optymalizacją

Jaki jest czas działania?

6. Proszę wykonać kopię procedury

paszynsk@atari:~/optimize/MM_dtime$cp mm5.c mm6.c

oraz dokonać w mm6.c kolejnej próby optymalizacji procedury MM poprzez umieszczenie wspólnie używanych zmiennych wewnątrze pętli w rejestrach

```
register double XXX;
```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

7. Proszę wykonać kopię procedury

paszynsk@atari:~/optimize/MM_dtime$cp mm6.c mm7.c

oraz dokonać w mm7.c kolejnej próby optymalizacji procedury MM poprzez zamianę wierszy i kolumn w macierzy second

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

8. Proszę wykonać kopię procedury

paszynsk@atari:~/optimize/MM_dtime$cp mm7.c mm8.c

Proszę dokonać blokowania operacji nie tylko w wierszach ale również w kolumnach

```
  for (i = 0; i < local_size; i++) {
    for (j = 0; j < local_size; j++) {
       for (k = 0; k < local_size; ) {
        if(j<SIZE-8 && k<SIZE-8) {
// TU PROSZE UZUPEŁNIĆ
```

```
            j=j+8;
            k=k+8;
          }
          else {
            sum = sum + first[i][k]*second[j][k];
            k++; j++;
        }
    }
}
```

Proszę skompilować procedurę z optymalizacją

Jaki jest czas działania?

9. 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.