Example: We have a code which is in the main file `multi.f90` and `subs.f90`. The compilation and linking of the code can be employed by `Makefile`:

```
FXX=gfortran

## optimize for computations
FFLAGS=  -fPIC -fdefault-real-8 -fopenmp -O2 -w -ffpe-trap=invalid,zero,overflow

# for debugging of the code
#FFLAGS= -fPIC -fdefault-real-8 -g -fbacktrace -fbounds-check -w -Wall -finit-real=nan -finit-integer=-999999
           -fno-align-commons -ffpe-trap=invalid,zero,overflow,denormal

all: multi

multi: multi.o  subs.o
$(FXX) $(FFLAGS) -o multi multi.o subs.o

multi.o: multi.f90
$(FXX) $(FFLAGS) -c multi.f90

subs.o: subs.f90
$(FXX) $(FFLAGS) -c subs.f90

clean:
rm -f multi *.o
```

Few options:

- `-fdefault-real-8` – all real variables are in double precision, some translators of fortran use `-r8`
- `-O2` – level of optimization
- `-W` – turn on warnings
- `-fPIC` – generate position-independent code (PIC) suitable for use in a shared library, recommended for `LAPCK`
- `-fbounds-check` – check the ranges of arrays
- `-fbacktrace` – if error is met, the sequence of callings is written

Matrices

$$\mathbb{A} = \{a_{ij}\}_{i,j=1}^{N}, \qquad a_{ij} = i + j, \quad i,j = 1,\dots,N,$$
$$\mathbb{B} = \{b_{ij}\}_{i,j=1}^{N}, \qquad b_{ij} = i - j, \quad i,j = 1,\dots,N,$$
$$\mathbb{C} = \{c_{ij}\}_{i,j=1}^{N}, \qquad \mathbb{C} = \mathbb{A}\mathbb{B} = ?$$

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{1,1} & \mathbb{A}_{1,2} & \dots & A_{1,M} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbb{A}_{M,1} & \mathbb{A}_{M,2} & \dots & A_{M,M} \end{pmatrix},$$
$$\mathbb{B} = \dots, \quad \mathbb{C} = \dots.$$

Using the code `multi.f90` and the `Makefile` from the link
`msekce.karlin.mff.cuni.cz/~dolejsi/Vyuka/NS_source/Cache/index.html`
test three variants of the matrix-matrix multiplications:

- simple multiplications per elements:

$$c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}, \qquad i,j = 1,\dots,N$$

- block multiplications with different block size n (the default size is $n = 40$):

$$\mathbb{C}_{ij} = \sum_{k=1}^{M} \mathbb{A}_{i,k}\mathbb{B}_{k,j}, \qquad i,j = 1,\dots,M, \quad M = \frac{N}{n}$$

- multiplications using Fortran90 function `matmul`.

```
C(1:N, 1:N) = matmul(A(1:N, 1:N), B(1:N, 1:N) )
```

Use different sizes of N. This codes measures the used computational time in seconds and gives the speed of computations in Mflops $= 2N^3/\text{time}/1E + 06$.

Basic tasks

1. Study the code line by line. If something is unclear, ask the teacher.
2. Find the "asymptotic" ratio of the speed of the computations among all 3 techniques. Hint: repeat the computations for increasing N till an almost constant speed is achieved.
3. Find the "optimal size" of the blocks for the block multiplications.
4. How does item 2. depend on the single/double precision? `-fPIC -fdefault-real-8`
5. How does item 2. depend on the optimization? `-O2`

Advanced task Let us consider matrices

$$\mathbb{A} = \{a_{ij}\}_{i,j=1}^{N}, \qquad a_{ij} = \begin{cases} i+j, & |i-j| \le 1 \\ 0 & \text{otherwise}, \end{cases}, i,j = 1,\dots,N,$$

$$\mathbb{B} = \{b_{ij}\}_{i,j=1}^{N}, \qquad b_{ij} = \begin{cases} i-j, & |i-j| \le 1 \\ 0 & \text{otherwise}, \end{cases}, i,j = 1,\dots,N,$$

$$\mathbb{C} = \{c_{ij}\}_{i,j=1}^{N}, \qquad \mathbb{C} = \mathbb{A}\mathbb{B} = ?$$

Matrices $\mathbb{A}$ and $\mathbb{B}$ are sparse, 3-diagonal?

1. What is the shape of matrix $\mathbb{C}$ (x-diagonal).
2. Write a code which avoids the multiplications by zeros.
   Hint: use the code `multi_sparse.f90` and modify `Makefile`.