



3rd Task Sheet

The tasks 1 and 2 are solved together during the tutorial. Task 3 is assigned as homework until the next tutorial date. Please hand in your solutions via OPAL or by e-mail.

Task 1

Consider the following fragment of an OpenMP programme. Decide at the code lines marked with (1) and (2) whether the variables are private or shared. Provide the values of the variables in the respective threads.

```
1
2  int i = 1;
3  int j = 2;
4
5  int f(int p)
6  {
7      p = 10;
8      printf("%d, %d, %d\n", i, j, p); /* (1) */
9  }
10
11 int main(void)
12 {
13     int k = 3;
14     int l = 4;
15     int o = 7;
16     int p = 8;
17
18     #pragma omp parallel shared(j) private(i,o) firstprivate(k)
19     {
20         int m = 5;
21         i = 6;
22         f(p);
23         printf("%d, %d, %d, %d, %d, %d, %d\n", i, j, k, l, m, o, p); /* (2) */
24     }
25 }
```

Task 2

Write a parallel programme with OpenMP and C which adds two vectors a and b of length n ($a, b \in \mathbb{R}^n$) with $c = a + b$. Use random values for initialising the vector elements.

Task 3

Compile the programme from task 2 and execute it. If you use the gcc compiler, you can switch on OpenMP by using the compiler option `-fopenmp`. The source code *vectoradd.c* is compiled to the binary *vectoradd* as follows:

```
gcc -fopenmp vectoradd.c -o vectoradd
```

Extend the source code by a sequential routine which also calculates $c = a + b$. Verify the result of the parallel routine by comparing it with the result obtained sequentially.