Perfomance Programming in C++

Exercise Sheet 02

Author: Juri Wiens

Exercise 5

`f` is a function that takes the following arguments:

- `v`: an integer value
- `r`: an integer reference
- `p`: a pointer to an integer value
- `pr`: a reference to an integer pointer

In general `f` has the following behavior: it prints the addresses and values of all arguments to stdout, as well as the dereferenced values of the pointers `p` and `pr`. After that it reassigns them.

Running `main` prints the following lines:

- Line 1:
  - Address of `v`: the address of `main`'s `v`
  - Address of `r`: equals to the address of `v`, because it is just an alias for `v`
  - Address of `p`: the address of `main`'s `p`
  - Value of `p`: `p` contains the address of `v`
  - Address of `pr`: equals to the address of `p`, because it is just an alias for `p`
  - Value of `pr`: equals to value of `p`
- Line 2:
  - Value of `v`: `0`, like it was initialized
  - Value of `r`: same as the value of `v`
  - Dereferenced value of `p`: equals to value of v, because it was initialized with the address of `v`
  - Dereferenced value of `pr`: same as the dereferenced value of `p`, because `pr` references `p`
- Line 3:
  - Address of `v`: the address of `f`'s copy of `v`
  - Address of `r`: same as on line 1, because `r` references `main`'s `v`
  - Address of `p`: the address of `f`'s copy of `p`
  - Value of `p`: equals to line 1, because it is a copy
  - Address of `pr`: same as on line 1, because `pr` references `main`'s `p`
  - Value of `pr`: same as on line 1, because it references `main`'s `p`
- Line 4:
  - Value of `v`: equals to line 2, because it is a copy of `main`'s `v`
  - Value of `r`: same as on line 2, because it references `main`'s `v`
  - Dereferenced value of `p`: same as on line 2, because `p` points to the same address
  - Dereferenced value of `pr`: same as on line 2, because `pr` references `main`'s `p`
- Line 5:
  - Address of `v`: same as on line 1, it is the unchanged address of `main`'s `v`
  - Address of `r`: same as on line 1 and 3
  - Address of `p`: same as on line 1, it is the unchanged address of `main`'s `p`
  - Value of `p`: equals to the address of `f`'s `v` on line 3, because it was reassigned to that value inside of `f` through the `pr` reference
  - Address of `pr`: same as on line 1 and 3
  - Value of `pr`: same as the value of `pr`, because it references `p`
- Line 6:
  - Value of `v`: `3`, because `f` reassigned this value via the `r` reference

– Value of `r`: same as the value of `v`, because `r` references `v`
– Dereferenced value of `p`: same as the value of `f`'s value of `v` on line 4 (see value of `p` on line 5)
– Dereferenced value of `pr`: same as the dereferenced value of `p`, because `pr` references `p`