

TP 2 - Problem 6 - I

```
1  #include <algorithm>
2  #include <cassert>
3  #include <iostream>
4  #include <vector>
5
6  #define SHOW(arg) std::cout << "Macro SHOW "" #arg "" : " << (arg) << '\n';
7
8  // Only to learn pointer arithmetic.
9  void reverseDoubles(double *dbl_ptr, int n) {
10     assert(n >= 0);
11     // "end_ptr" points to the element following the last one.
12     double *end_ptr{dbl_ptr + n};
13     n /= 2;
14     while (--n >= 0) {
15         double tmp{*dbl_ptr};
16         // Post-increment "dbl_ptr" but pre-decrement "end_ptr"...
17         *dbl_ptr++ = *--end_ptr;
18         *end_ptr = tmp;
19     }
```

TP 2 - Problem 6 - II

```
20 }
21 // The right version using the standard library.
22 void reverseDoublesSTL(std::vector<double> &v) { std::ranges::reverse(v); }
23
24 int main() {
25     {
26         double a[]{};
27         reverseDoubles(a, 0);
28         std::vector<double> v{};
29         reverseDoublesSTL(v);
30     }
31     {
32         double a[]{1, 2};
33         reverseDoubles(a, 2);
34         SHOW(a[0])
35         SHOW(a[1])
36         std::vector<double> v{1, 2};
37         reverseDoublesSTL(v);
38         SHOW(v[0])
39         SHOW(v[1])
```

TP 2 - Problem 6 - III

```
40     }
41     {
42         double a[]{1, 2, 3};
43         reverseDoubles(a, 3);
44         SHOW(a[0])
45         SHOW(a[1])
46         SHOW(a[2])
47         std::vector<double> v{1, 2, 3};
48         reverseDoublesSTL(v);
49         SHOW(v[0])
50         SHOW(v[1])
51         SHOW(v[2])
52     }
53     {
54         double a[]{1, 2, 3, 4};
55         reverseDoubles(a, 4);
56         SHOW(a[0])
57         SHOW(a[1])
58         SHOW(a[2])
59         SHOW(a[3])
```

TP 2 - Problem 6 - IV

```
60     std::vector<double> v{1, 2, 3, 4};
61     reverseDoublesSTL(v);
62     SHOW(v[0])
63     SHOW(v[1])
64     SHOW(v[2])
65     SHOW(v[3])
66 }
67 return 0;
68 }
```

TP 2 - Problem 6 - V

Output:

```
1 Macro SHOW "a[0]": 2
2 Macro SHOW "a[1]": 1
3 Macro SHOW "v[0]": 2
4 Macro SHOW "v[1]": 1
5 Macro SHOW "a[0]": 3
6 Macro SHOW "a[1]": 2
7 Macro SHOW "a[2]": 1
8 Macro SHOW "v[0]": 3
9 Macro SHOW "v[1]": 2
10 Macro SHOW "v[2]": 1
11 Macro SHOW "a[0]": 4
12 Macro SHOW "a[1]": 3
13 Macro SHOW "a[2]": 2
14 Macro SHOW "a[3]": 1
15 Macro SHOW "v[0]": 4
16 Macro SHOW "v[1]": 3
17 Macro SHOW "v[2]": 2
18 Macro SHOW "v[3]": 1
```