

```
1 # 8-pointer
2
3 ## `radius.c`
4
5 ### On Variables
6 - type, value, address
7 - `&`: address-of operator
8 - printf the address (`%p`)
9 - `lvalue`, `rvalue`???
10
11 ### On Pointers
12 - `int *` syntax
13 - int * vs. double * (type cast???)
14 - refs to itself (int ** vs. int *)
15 - Visualization
16
17 - `scanf`: how does it work???
18
19 ## `Swap` (`selection-sort.c`)
20 - `WrongSwap`
21 - `Swap`
22 - Visualization
23
24 ## Pointers and Arrays (`selection-sort.c`)
25
26 - `()`: function call operator
27 - `SelectionSort(numbers, LEN)`
28 - `int arr[]` vs. `(int *arr)`
29 - `numbers[i]` vs. `*(numbers + i)`
30 - pointers arithmetic (in arrays!!!)
31 - `pointer + int`, `pointer - int`, `pointer - pointer`
32 - `&numbers[i]` vs. `numbers + i`
33
34 ## Array Name (`selection-sort.c`)
35 - `int arr[] = {1, 2, 3};`
36 - `arr++`
37 - `numbers++`
38
39 ## Dynamic Memory Management (`selection-sort.c`)
40
41 - VLA
42 - `malloc.h` vs. `stdlib.h`
43 - `malloc`
44 - `void *`
45   - `int *`
46   - `sizeof(*numbers)`
47 - size = 0: implementation-defined
48 - `unsigned long long`
49 - `NULL`
50 - `(void *) 0`
51 - `free`
52 - memory leak (heap)
53 - **undefined behaviors**
```

```
54     - double `free`  
55     - `free` non-`malloc`  
56     - `numbers = NULL`  
57     - dereference `free`d memory  
58  
59 ## `const` in `Print` (`selection-sort.c`)
```