

```

1 //
2 // Created by hfwei on 2023/10/12.
3 // Visualization of Swap: https://pythontutor.com/visualize.html#code
  =//%0A//%20Created%20by%20hfwei%20on%202023/10/12.%0A//%0A%0A%
  23include%20%3Cstdio.h%3E%0A%0A%23define%20LEN%205%0A%0Aavoid%
  20SelectionSort%28int%20arr%5B%5D,%20int%20len%29%3B%0Aavoid%
  20WrongSwap%28int%20left,%20int%20right%29%3B%0Aavoid%20Swap%28int%20*
  left,%20int%20*right%29%3B%0Aint%20GetMinIndex%28const%20int%20arr%5B%
  5D,%20int%20begin,%20int%20end%29%3B%0Aavoid%20Print%28const%20int%
  20arr%5B%5D,%20int%20len%29%3B%0A%0Aint%20main%28void%29%20%7B%0A%20%
  20int%20numbers%5BLEN%5D%20%3D%20%7B15,%2078,%2023,%208,%2050%7D%3B%0A
  %0A%20%20Print%28numbers,%20LEN%29%3B%0A%20%20SelectionSort%28numbers
  ,%20LEN%29%3B%0A%20%20Print%28numbers,%20LEN%29%3B%0A%0A%20%20return%
  200%3B%0A%7D%0A%0A//%20arr%3A%20the%20%28copy%20of%20the%29%20address%
  20of%20the%20first%20element%20of%20the%20%60numbers%60%20array%0Aavoid
  %20SelectionSort%28int%20arr%5B%5D,%20int%20len%29%20%7B%0A%20%20for%
  20%28int%20i%20%3D%200%3B%20i%20%3C%20len%3B%20i%2B%2B%29%20%7B%0A%20%
  20%20%20int%20min_index%20%3D%20GetMinIndex%28arr,%20i,%20len%29%3B%0A
  %0A%20%20%20%20//%20ERROR%3A%20WrongSwap%28arr%5B%5D,%20arr%
  5Bmin_index%5D%29%3B%0A%20%20%20%20int%20temp%20%3D%20arr%5B%5D%3B%0A
  %20%20%20%20arr%5B%5D%20%3D%20arr%5Bmin_index%5D%3B%0A%20%20%20%20arr
  %5Bmin_index%5D%20%3D%20temp%3B%0A%20%20%7D%0A%7D%0A%0Aint%
  20GetMinIndex%28const%20int%20arr%5B%5D,%20int%20begin,%20int%20end%29
  %20%7B%0A%20%20int%20min%20%3D%20arr%5Bbegin%5D%3B%0A%20%20int%
  20min_index%20%3D%20begin%3B%0A%0A%20%20for%20%28int%20i%20%3D%20begin
  %20%2B%201%3B%20i%20%3C%20end%3B%20%2B%2B%29%20%7B%0A%20%20%20%20if%
  20%28arr%5B%5D%20%3C%20min%29%20%7B%0A%20%20%20%20%20min%20%3D%
  20arr%5B%5D%3B%0A%20%20%20%20%20min_index%20%3D%20i%3B%0A%20%20%20
  %20%7D%0A%20%20%7D%0A%0A%20%20return%20min_index%3B%0A%7D%0A%0Aavoid%
  20WrongSwap%28int%20left,%20int%20right%29%20%7B%0A%20%20int%20temp%20
  %3D%20left%3B%0A%20%20left%20%3D%20right%3B%0A%20%20right%20%3D%20temp
  %3B%0A%7D%0A%0Aavoid%20Swap%28int%20*left,%20int%20*right%29%20%7B%0A%
  20%20int%20temp%20%3D%20*left%3B%0A%20%20*left%20%3D%20*right%3B%0A%20
  %20*right%20%3D%20temp%3B%0A%7D%0A%0Aavoid%20Print%28const%20int%20arr%
  5B%5D,%20int%20len%29%20%7B%0A%20%20printf%28%22%5Cn%22%29%3B%0A%20%
  20for%20%28int%20i%20%3D%200%3B%20i%20%3C%20len%3B%20i%2B%2B%29%20%7B%
  0A%20%20%20%20printf%28%22%25d%20%22,%20arr%5B%5D%29%3B%0A%20%20%7D%
  0A%20%20printf%28%22%5Cn%22%29%3B%0A%7D&cumulative=true&heapPrimitives
  =nevernest&mode=edit&origin=opt-frontend.js&py=c_gcc9.3.0&
  rawInputLstJSON=%5B%5D&textReferences=false
4 // Visualization of malloc:
5 //
6
7 #include <stdio.h>
8 #include <stdlib.h>
9
10 // #define LEN 5
11
12 void SelectionSort(int arr[], int len);
13 void WrongSwap(int left, int right);
14 void Swap(int *left, int *right);
15 int GetMinIndex(const int arr[], int begin, int end);
16 void Print(const int arr[], int len);

```

```

17
18 int main(void) {
19     // int numbers[LEN] = {15, 78, 23, 8, 50};
20
21     int len = 0;
22     scanf("%d", &len);
23
24     // VLA
25     // int numbers[len];
26
27     // return value: (void *) 0
28     int *numbers = malloc(len * sizeof(*numbers));
29
30     // NULL: null pointer (void *) 0
31     if (numbers == NULL) {
32         printf("Memory allocation failed!\n");
33         return 0;
34     }
35
36     for (int i = 0; i < len; i++) {
37         scanf("%d", &numbers[i]);
38     }
39
40     Print(numbers, len);
41     // &numbers[0] (numbers[0] is also a variable) of type (int *)
42     SelectionSort(numbers, len);
43     Print(numbers, len);
44
45     return 0;
46 }
47
48 // arr: the (copy of the) address of the first element of the `numbers
   // array
49 // int arr[] <-> int *arr (in compilers)
50 void SelectionSort(int arr[], int len) {
51     for (int i = 0; i < len; i++) {
52         int min_index = GetMinIndex(arr, i, len);
53
54         // ERROR: WrongSwap(arr[i], arr[min_index]);
55         // int temp = arr[i];
56         // arr[i] = arr[min_index];
57         // arr[min_index] = temp;
58
59         // &arr[i] <=> &*(arr + i) <=> arr + i
60         Swap(&arr[i], &arr[min_index]);
61     }
62 }
63
64 // int arr[] <-> int *arr (in compilers)
65 int GetMinIndex(const int arr[], int begin, int end) {
66     int min = arr[begin];
67     int min_index = begin;
68

```

```
69     for (int i = begin + 1; i < end; ++i) {
70         // arr[i] <-> *(arr + i) <-> *(i + arr) <-> i[arr] (subscript
           operator)
71         if (arr[i] < min) {
72             min = arr[i];
73             min_index = i;
74         }
75     }
76
77     return min_index;
78 }
79
80 void WrongSwap(int left, int right) {
81     int temp = left;
82     left = right;
83     right = temp;
84 }
85
86 void Swap(int *left, int *right) {
87     int temp = *left;
88     *left = *right;
89     *right = temp;
90 }
91
92 void Print(const int arr[], int len) {
93     printf("\n");
94     for (int i = 0; i < len; i++) {
95         printf("%d ", arr[i]);
96     }
97     printf("\n");
98 }
```