

```
1 // Created by hfwei on 2024/11/15.
2
3 #include <stdio.h>
4
5 int main(void) {
6     int i = 50;
7
8     scanf("%d", &i);
9     printf("%d\n", i);
10
11     return 0;
12 }
```

```

1 // Created by hengxin on 11/06/24.
2 // pythontutor:
3 // https://pythontutor.com/render.html#code=%23include%20%
3Cstdio.h%3E%0A%0A%23define%20PI%203.14%0A%0Aint%20main%
28void%29%20%7B%0A%20%20/%20https%3A//intellij-support.
jetbrains.com/hc/en-us/community/posts/115000740490-Where-
did-the-black-windows-go-%3Fpage%3D1%
23community_comment_115000619510%0A%20%20/%20%20setbuf%
28stdout,%20NULL%29%3B%0A%0A%20%20/*****%200n%
20radius_1%20*****/%0A%20%20/%20type%3A%20int%3B%
20value%3A%20100%3B%20address%3A%20%26radius_1%20%28%26%3A
%20address-of%20operator%29%0A%20%20int%20radius_1%20%3D%
20100%3B%0A%0A%20%20printf%28%22radius_1%20%3D%20%25d%5Cn%
22,%20radius_1%29%3B%0A%20%20printf%28%22%26radius_1%20%3D
%20%25p%5Cn%22,%20%26radius_1%29%3B%0A%0A%20%20/%20lvalue
%20and%20rvalue%0A%20%20radius_1%20%3D%20200%3B%0A%20%20
/%20lvalue%20conversion%20%3D%3E%20rvalue%0A%20%20double%
20circumference%20%3D%202%20*%20PI%20*%20radius_1%3B%0A%20
%20printf%28%22circumference%20%3D%20%25f%5Cn%22,%
20circumference%29%3B%0A%20%20/*****%200n%20radius_1%
20*****/%0A%0A%20%20/*****%200n%20ptr_radius_1%
20*****/%0A%20%20/%20type%3A%20int%20*%3B%20value%3A
%20%26radius_1%0A%20%20int%20*ptr_radius_1%20%3D%20%
26radius_1%3B%0A%0A%20%20printf%28%22%26ptr_radius_1%20%3D
%20%25p%5Cn%22,%20%26ptr_radius_1%29%3B%0A%20%20
/*****%200n%20ptr_radius_1%20*****/%0A%0A%20%20
/*****%200n%20ptr_radius_1%20as%20lvalue%20and%
20rvalue%20*****/%0A%20%20int%20radius_2%20%3D%201000
%3B%0A%20%20int%20*ptr_radius_2%20%3D%20%26radius_2%3B%0A%
0A%20%20ptr_radius_1%20%3D%20ptr_radius_2%3B%0A%20%20*
ptr_radius_2%20%3D%202000%3B%0A%20%20printf%28%22radius_1%
20%3D%20%25d%5Cn%22,%20*ptr_radius_1%29%3B%0A%20%20
/*****%200n%20ptr_radius_1%20as%20lvalue%20and%
20rvalue%20*****/%0A%0A%20%20/*****%200n%20*
ptr_radius_1%20*****/%0A%20%20/%20*%3A%20indirection
/dereference%20operator%0A%20%20/%20*ptr_radius_1%
20behaves%20like%20radius_1%0A%20%20*ptr_radius_1%20%3D%
20200%3B%0A%20%20printf%28%22radius_1%20%3D%20%25d%5Cn%22
,%20radius_1%29%3B%0A%20%20circumference%20%3D%202%20*%
20PI%20*%20%28*ptr_radius_1%29%3B%0A%20%20/*****%200n
%20*ptr_radius_1%20*****/%0A%0A%20%20int%20v%20%3D%
20100%3B%0A%20%20int%20*pv%20%3D%20%26v%3B%0A%20%20printf%
28%22pv%20%3A%20%25p%5Cn%20*pv%20%3A%20%25d%5Cn%22,%20pv,%
20*pv%29%3B%0A%20%20pv%20%3D%20%26pv%3B%0A%20%20printf%28%

```

```

3 22pv%20%3A%20%25p%5Cn%22,%20pv%29%3B%0A%0A%20%20return%200
  %3B%0A%7D&cumulative=true&curInstr=0&heapPrimitives=
  nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.
  0&rawInputLstJSON=%5B%5D&textReferences=false
4
5 #include <stdio.h>
6
7 #define PI 3.14
8
9 int main(void) {
10 // https://intellij-support.jetbrains.com/hc/en-us/
  community/posts/115000740490-Where-did-the-black-windows-
  go-?page=1#community_comment_115000619510
11 // setbuf(stdout, NULL);
12
13 /***** On radius_1 as lvalue and rvalue *****/
14 int radius_1 = 100;
15
16 printf("radius_1 = %d\n", radius_1);
17
18 double circumference = 2 * PI * radius_1;
19 printf("circumference = %f\n", circumference);
20 /***** On radius_1 as lvalue and rvalue *****/
21
22 /***** On ptr_radius_1 *****/
23 /***** On ptr_radius_1 *****/
24
25 /***** On ptr_radius_1 as lvalue and rvalue
  *****/
26 /***** On ptr_radius_1 as lvalue and rvalue
  *****/
27
28 /***** On *ptr_radius_1 *****/
29 /***** On *ptr_radius_1 *****/
30
31 /***** On types of pointers *****/
32 /***** On types of pointers *****/
33
34 /***** On types of pointers (more) *****/
35 /***** On types of pointers (more) *****/
36
37 return 0;
38 }

```

```
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`)
```

```
1 add_executable(pointer pointer.c)
2 add_executable(selection-sort-pointers selection-sort.c)
3 add_executable(pointer-array pointer-array.c)
4 add_executable(pointer-const pointer-const.c)
```

```
1 // Created by hfwei on 2024/11/6.
2
3 int main(void) {
4     /***** On array name *****/
5     /***** On array name *****/
6
7     /***** On malloc/free *****/
8     /***** On malloc/free *****/
9
10    return 0;
11 }
```

```

1 // Created by hfwei on 2024/11/6.
2 // Python tutor:
3 // https://pythontutor.com/render.html#code=%23include%20%
3Cstdio.h%3E%0A%0Aint%20main%28void%29%20%7B%0A%20%20/%
20v%3A%20int,%20const%20int%0A%20%20/%20pv%3A%20int%20*,%
20int%20*%20const%0A%20%20/%20const%20int%20*,%20const%
20int%20*%20const,%20int%20const%20*,%20int%20const%20*%
20const%0A%0A%20%20int%20var%20%3D%200%3B%0A%0A%20%20/%
20int%20var_1%0A%20%20int%20var_1%20%3D%2010%3B%0A%0A%20%
20int%20*ptr_1%20%3D%20%26var_1%3B%0A%20%20*ptr_1%20%3D%
20%20%3B%0A%20%20printf%28%22var_1%20%3D%20%25d%5Cn%22,%
20var_1%29%3B%0A%20%20ptr_1%20%3D%20%26var%3B%0A%0A%20%
20const%20int%20*ptr_1_1%20%3D%20%26var_1%3B%0A%20%20/%20
%20*ptr_1_1%20%3D%2030%3B%20%20/%20Wrong%0A%20%20printf%
28%22var_1%20%3D%20%25d%5Cn%22,%20var_1%29%3B%0A%20%
20ptr_1%20%3D%20%26var%3B%0A%0A%20%20int%20*const%
20ptr_1_2%20%3D%20%26var_1%3B%0A%20%20*ptr_1_2%20%3D%2040%
3B%0A%20%20printf%28%22var_1%20%3D%20%25d%5Cn%22,%20var_1%
29%3B%0A%20%20/%20%20*ptr_1_2%20%3D%20%26var%3B%20%20/%
20Wrong%0A%0A%20%20/%20const%20int%20var_2%0A%20%20const%
20int%20var_2%20%3D%20100%3B%0A%0A%20%20int%20*ptr_2%20%3D
%20%26var_2%3B%0A%20%20*ptr_2%20%3D%20200%3B%0A%20%
20printf%28%22var_2%20%3D%20%25d%5Cn%22,%20var_2%29%3B%0A%
0A%20%20const%20int%20*ptr_2_1%20%3D%20%26var_2%3B%0A%20%
20/%20%20*ptr_2_1%20%3D%20300%3B%0A%20%20printf%28%
22var_2%20%3D%20%25d%5Cn%22,%20var_2%29%3B%0A%20%20const%
20int%20*const%20ptr_2_2%20%3D%20%26var_2%3B%0A%20%20/%20
%20*ptr_2_2%20%3D%20400%3B%0A%20%20printf%28%22var_2%20%3D
%20%25d%5Cn%22,%20var_2%29%3B%0A%0A%20%20return%200%3B%0A%
7D&cumulative=true&curInstr=0&heapPrimitives=nevernest&
mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&
rawInputLstJSON=%5B%5D&textReferences=false
4
5 #include <stdio.h>
6
7 int main(void) { return 0; }

```



```

1 // Created by hfwei on 2024/11/06.
2 // Visualization of Swap:
3 // https://pythontutor.com/render.html#code=//%20Created%
  20by%20hfwei%20on%202024/10/12.%0A%0A%0A%23include%20%
  3Cstdio.h%3E%0A%23include%20%3Cstdlib.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%7B%2025,%
  2078,%2015,%2023,%2011%20%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%0Aavoid%20SelectionSort%28int%20*arr,%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
  %20%20%20%20Swap%28arr%20%2B%20i,%20arr%20%2B%20min_index%
  29%3B%0A%20%20%7D%0A%7D%0A%0Aint%20GetMinIndex%28const%
  20int%20*arr,%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%2Bi%29
  %20%7B%0A%20%20%20%20if%20%28arr%5Bi%5D%20%3C%20min%29%20%
  7B%0A%20%20%20%20%20min%20%3D%20arr%5Bi%5D%3B%0A%20%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%5Bi%5D%29%3B%0A%20%20%7D%0A%20%
  20printf%28%22%5Cn%22%29%3B%0A%7D&cumulative=true&curInstr
  =0&heapPrimitives=nevernest&mode=display&origin=opt-
  frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&
  textReferences=false
4 // Visualization of malloc:

```

```
5
6 #include <stdio.h>
7 #include <stdlib.h>
8
9 #define LEN 5
10
11 void SelectionSort(int arr[], int len);
12 int GetMinIndex(const int arr[], int begin, int end);
13 void WrongSwap(int left, int right);
14 void Swap(int arr[], int left_index, int right_index);
15 void Print(const int arr[], int len);
16
17 int main(void) {
18     int numbers[LEN] = {5, 2, 8, 1, 9};
19
20     Print(numbers, LEN);
21     SelectionSort(numbers, LEN);
22     Print(numbers, LEN);
23
24     return 0;
25 }
26
27 void SelectionSort(int *arr, int len) {
28     for (int i = 0; i < len; i++) {
29         int min_index = GetMinIndex(arr, i, len);
30         Swap(arr, i, min_index);
31         // WrongSwap(arr[i], arr[min_index]);
32     }
33 }
34
35 int GetMinIndex(const int arr[], int begin, int end) {
36     int min = arr[begin];
37     int min_index = begin;
38
39     for (int i = begin + 1; i < end; ++i) {
40         if (arr[i] < min) {
41             min = arr[i];
42             min_index = i;
43         }
44     }
45
46     return min_index;
47 }
48
```

```
49 void WrongSwap(int left, int right) {
50     int temp = left;
51     left = right;
52     right = temp;
53 }
54
55 void Swap(int arr[], int left_index, int right_index) {
56     int temp = arr[left_index];
57     arr[left_index] = arr[right_index];
58     arr[right_index] = temp;
59 }
60
61 void Print(const int arr[], int len) {
62     printf("\n");
63     for (int i = 0; i < len; i++) {
64         printf("%d ", arr[i]);
65     }
66     printf("\n");
67 }
```