```c
/**
 * file: min-max.c
 *
 * Created by hengxin on 11/28/21.
 */

#include <stdio.h>
#define LEN 20
int numbers[LEN] = {0};

void MinMax(const int nums[], int len, int *min, int *max);

int main() {
  int len = -1;
  while (scanf("%d", &numbers[++len]) != EOF);

  int min = 0;
  int max = 0;
  MinMax(numbers, len, &min, &max);

  printf("min = %d, max = %d\n", min, max);

  return 0;
}

void MinMax(const int nums[], int len, int *min, int *max) {
  *min = nums[0];
  *max = nums[0];

  for (int i = 1; i < len; i++) {
    if (nums[i] < *min) {
      *min = nums[i];
    }

    if (nums[i] > *max) {
      *max = nums[i];
    }
  }
}

```

```c
/**
 * file: pointer.c
 *
 * Created by hengxin on 11/28/21.
 */

#include <stdio.h>

int main() {
  int radius = 10;

  printf("radius = %d\n", radius);
  printf("&radius = %p\n", &radius);

  double circumference = 2 * 3.14 * radius;
  radius = 20;
  printf("radius = %d; circumference = %f\n", radius, circumference);
  printf("&radius = %p\n", &radius);

  int *ptr_radius = &radius;
  circumference = 2 * 3.14 * (*ptr_radius);
  *ptr_radius = 30;

  printf("radius = %d; circumference = %f\n", *ptr_radius, circumference);
  printf("&radius = %p\n", &radius);

  int radius_2 = 100;
  int *ptr_radius_2 = &radius_2;

  ptr_radius = ptr_radius_2;
  printf("radius = %d\n", *ptr_radius);

  *ptr_radius_2 = 200;
  printf("radius = %d\n", *ptr_radius);

  /**
   * Unfortunately, the compiler only complains about this!!!
   */
  const int radius_3 = 1000;
  ptr_radius = &radius_3;
  *ptr_radius = 2000;
  printf("radius = %d\n", *ptr_radius);
  printf("radius = %d\n", radius_3);

  int arr[5] = {0};
  int *ptr_array = arr;
  ptr_array++;

  return 0;
}
```

```c
/**
 * file: selection-sort.c
 *
 * Created by hengxin on 11/28/21.
 */

#include <stdio.h>
#include <stdlib.h>

void Swap(int *left, int *right);
void Print(const int arr[], int len);

/**
 * Sort the array ARR of length LEN using the selection sort algorithm.
 *
 * @param arr The array to be sorted.
 * @param len The length of the array.
 */
//void SelectionSort(int arr[], int len);
void SelectionSort(int *arr, int len);

int main() {
  /**
   * Input the array
   * Note: fails to run this program in "Run" (Ctrl + D)
   * See: https://youtrack.jetbrains.com/issue/CPP-5704
   * Use "Terminal" instead.
   */
  int len = 0;
  printf("Please enter the length of the array to sort.\n");
  scanf("%d", &len);
  int *numbers = malloc(len * sizeof *numbers);

  printf("Please enter %d integers.\n", len);
  if (numbers == NULL) {
    printf("Error! Memory Not Allocated!\n");
    return 0;
  }

  for (int i = 0; i < len; i++) {
//    scanf("%d", &numbers[i]);
    scanf("%d", numbers + i);
  }

  SelectionSort(numbers, len);
  Print(numbers, len);

  free(numbers);

  return 0;
}

void Print(const int arr[], int len) {
  printf("\n");
  for (int i = 0; i < len; i++) {
    printf("%d ", arr[i]);
  }
  printf("\n");
}

void SelectionSort(int *arr, int len) {
  for (int i = 0; i < len; ++i) {
    int min = arr[i];
//    int min = i[arr]; // DO NOT DO THIS UGLY TRICK!!!
    int min_index = i;

    for (int j = i + 1; j < len; j++) {
      if (min > arr[j]) {
        min = arr[j];
        min_index = j;
      }
    }

    Swap(&arr[i], &arr[min_index]);
  }
}

void Swap(int *left, int *right) {
  int tmp = *left;
  *left = *right;
  *right = tmp;
}
```