

# **Examples using if-else**

# Outline

- **Example 1:** Computing the square root
- **Example 2:** Finding the number of real roots of a quadratic equation.
- **Example 3:** Outputting three numbers in a non-descending order.

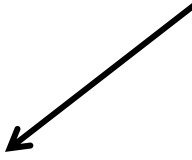
# Example 1

- **Objective:** To write a program to output the square root of a number.
- This example illustrates also how to use a pre-defined function to compute the square root of a number.

# Example 1: Solution

```
1  #include <stdio.h>
2  #include <math.h>           // You need this line to use sqrt()
3
4  int main(void) {
5      double num, result;
6
7      scanf("%lf", &num);
8
9      if (num >= 0) {
10         result = sqrt(num);
11         printf("The square root of %.4f is %.4f.\n", num, result);
12     }
13     else
14         printf("Can't compute square root for negative number.\n");
15
16     return 0;
17 }
18
```

**sqrt(num)** computes the square root of num and yields the result as a value of type double.



## Example 2

- **Objective:** Write the code segment to output the number of real number solutions of a quadratic equation in the form  $ax^2 + bx + c = 0$ . The code will read  $a$ ,  $b$ , and  $c$  from the user. We assume  $a \neq 0$ .
- **Approach:**
  - Compute discriminant as  $b^2 - 4ac$
  - discriminant  $> 0 \rightarrow$  2 real number solutions
  - discriminant  $= 0 \rightarrow$  1 real number solution
  - discriminant  $< 0 \rightarrow$  0 real number solutions

# Example 2: Solution #1

```
1 double a, b, c;           // To store the coefficients
2 scanf("%lf%lf%lf", &a, &b, &c);
3
4 if (b * b - 4 * a * c > 0)
5     printf("# of real number solutions: 2\n");
6
7 if (b * b - 4 * a * c == 0)
8     printf("# of real number solutions: 1\n");
9
10 if (b * b - 4 * a * c < 0)
11     printf("# of real number solutions: 0\n");
12
13
14
```

## Example 2: Solution #2

```
1 double a, b, c;           // The coefficients
2 double dis;               // The discriminant
3 int sol;                  // # of real number solutions
4
5 scanf("%lf%lf%lf", &a, &b, &c);
6 dis = b * b - 4 * a * c;  // Compute discriminant only once
7
8 if (dis > 0)
9     sol = 2;
10 else if (dis == 0)
11     sol = 1;
12 else                          // Otherwise dis < 0
13     sol = 0;
14
15 printf("# of real number solutions: %d\n", sol);
16
```

# Notes about Example 2

- In solution #2,  $b*b-4*a*c$  is only evaluated once, and thus the amount of computation is reduced.
- Solution #2 uses one `printf()` to output the result. The advantage is, if we need to change the output format, we only need to change one `printf()` statement.



# Example 3

- **Objective:** Write the code segment to read three integers from a user and output them in a non-descending order.
  - Assume the values are stored in variables **x**, **y**, and **z**
- Approach #1:
  - For each of the six possible arrangements, output the result accordingly: (1) **x** ≤ **y** ≤ **z**, (2) **x** ≤ **z** ≤ **y**, (3) **y** ≤ **x** ≤ **z**, (4) **y** ≤ **z** ≤ **x**, (5) **z** ≤ **x** ≤ **y**, (6) **z** ≤ **y** ≤ **x**
- Approach #2:
  - Sort the values of **x**, **y**, and **z** so that **x** ≤ **y** ≤ **z**.

# Example 3: Solution #1a

```
1  int x, y, z;                // To store input values
2  scanf("%d%d%d", &x, &y, &z);
3
4  if (x <= y && y <= z)
5      printf("%d %d %d\n", x, y, z);
6  else if (x <= z && z <= y)
7      printf("%d %d %d\n", x, z, y);
8  ...
9  else if (z <= x && x <= y)
10     printf("%d %d %d\n", z, x, y);
11 else
12     printf("%d %d %d\n", z, y, x);
13
14
15
16
```

## Example 3: Solution #1b (Broken implementation)

```
1  int x, y, z;                // To store input values
2  scanf("%d%d%d", &x, &y, &z);
3
4  if (x <= y && y <= z)
5      printf("%d %d %d\n", x, y, z);
6  if (x <= z && z <= y)
7      printf("%d %d %d\n", x, z, y);
8  if (y <= x && x <= z)
9      printf("%d %d %d\n", y, x, z);
10 if (y <= z && z <= x)
11     printf("%d %d %d\n", y, z, x);
12 if (z <= x && x <= y)
13     printf("%d %d %d\n", z, x, y);
14 if (z <= y && y <= x)
15     printf("%d %d %d\n", z, y, x);
16
```

Without using **else**, this implementation would produce multiple output when two or more input variables share the same value.

e.g., when **x, y, z** are all 6, the output "6 6 6" would appear six times.

## Example 3: Solution #2

```
1  int x, y, z, tmp;
2  scanf("%d%d%d", &x, &y, &z);
3
4  // First, make sure that x holds the smallest value
5  if (x > y) {                      // If y is smaller
6      tmp = x; x = y; y = tmp;      // swap x and y
7  }
8  if (x > z) {                      // If z is even smaller
9      tmp = x; x = z; z = tmp;      // swap x and z
10 }
11
12 // Next, make sure y <= z
13 if (y > z) {                      // If z is smaller
14     tmp = y; y = z; z = tmp;      // swap y and z
15 }
16
17 printf("%d %d %d\n", x, y, z);
```