



Degree in Software Engineering – Computing Basics

Solution to Unit 2.4 Exercises: Control Structures

This document includes the solution to the exercises of the document “Unit 2.4 Exercises: Control Structures”. It is recommended that you try doing the exercises without looking at the solutions first, and then you check your answers. Please note there might be multiple solutions to the same problem.

Exercise 1

Proposed solution:

```
flp = float(input("Pressure of the front-left tire (bar): "))
frp = float(input("Pressure of the front-right tire (bar): "))
blp = float(input("Pressure of the back-left tire (bar): "))
brp = float(input("Pressure of the back-right tire (bar): "))

pressure = 2.5
margin = 0.2

if flp < pressure - margin:
    print("The FRONT-LEFT tire has LOW PRESSURE")
elif flp > pressure + margin:
    print("The FRONT-LEFT tire has HIGH PRESSURE")
if frp < pressure - margin:
    print("The FRONT-RIGHT tire has LOW PRESSURE")
elif frp > pressure + margin:
    print("The FRONT-RIGHT tire has HIGH PRESSURE")
if blp < pressure - margin:
    print("The BACK-LEFT tire has LOW PRESSURE")
elif blp > pressure + margin:
    print("The BACK-LEFT tire has HIGH PRESSURE")
if brp < pressure - margin:
    print("The BACK-RIGHT tire has LOW PRESSURE")
elif brp > pressure + margin:
    print("The BACK-RIGHT tire has HIGH PRESSURE")
```

Exercise 2

Proposed solution:

```
a = float(input("Length of the first side: "))
b = float(input("Length of the second side: "))
c = float(input("Length of the third side: "))

if a + b > c and a + c > b and b + c > a:
    print("The triangle exists")
else:
    print("The triangle does NOT exist")
```



Exercise 3

Proposed solution:

```
score1 = int(input("Player 1 score: "))
score2 = int(input("Player 2 score: "))

if score1 > 21 and score2 > 21:
    print("Player 1 and player 2 lost!")
elif score1 <= 21 and (score2 > 21 or score1 > score2):
    print("Player 1 wins with {} points!".format(score1))
elif score2 <= 21 and (score1 > 21 or score2 > score1):
    print("Player 2 wins with {} points!".format(score2))
else:
    print("Player 1 and player 2 tied with {} points!".format(score1))
```

Exercise 4

Proposed solution:

```
score1 = 0
score2 = 0

finished = False
while score1 <= 21 and not finished:
    points = int(input("Player 1 ({} points): ".format(score1)))
    if points > 11:
        print("{} is not a valid score".format(points))
    elif points > 0:
        score1 = score1 + points
    else:
        finished = True
print("Player 1 ends their turn with {} points".format(score1))

finished = False
while score2 <= 21 and not finished:
    points = int(input("Player 2 ({} points): ".format(score2)))
    if points > 11:
        print("{} is not a valid score".format(points))
    elif points > 0:
        score2 = score2 + points
    else:
        finished = True
print("Player 2 ends their turn with {} points".format(score2))

if score1 > 21 and score2 > 21:
    print("Player 1 and player 2 lost!")
elif score1 <= 21 and (score2 > 21 or score1 > score2):
    print("Player 1 wins with {} points!".format(score1))
elif score2 <= 21 and (score1 > 21 or score2 > score1):
    print("Player 2 wins with {} points!".format(score2))
else:
    print("Player 1 and player 2 tied with {} points!".format(score1))
```



Exercise 5

Proposed solution:

```
accum = 0.0
count = 0

finished = False
while not finished:
    n = float(input("Next number: "))
    if n != 0:
        accum = accum + n
        count = count + 1
    else:
        finished = True

print("The arithmetic mean is {:.2f}".format(accum / count))
```

Exercise 6

Proposed solution:

```
e_last = 0.0
e_curr = 0.0
count = 0
factorial = 1

threshold = float(input("Specify the threshold to calculate e: "))
while threshold <= 0.0 or threshold > 1.0:
    print("Threshold must be in the interval (0.0, 1.0]")
    threshold = float(input("Specify the threshold to calculate e: "))

finished = False
while not finished:
    e_last = e_curr
    e_curr = e_curr + 1 / factorial
    count = count + 1
    factorial = factorial * count

    if e_curr - e_last <= threshold:
        finished = True

print("The approximation of e after {} terms is {}".format(count, e_curr))
```



Exercise 7

Proposed solution:

```
e = 0.0
count = 0
factorial = 1

iterations = int(input("Specify the number of iterations to calculate e: "))
while iterations <= 0:
    print("The number of iterations must be greater than 0")
    iterations = int(input("Specify the number of iterations to calculate e: "))

for i in range(iterations):
    e = e + 1 / factorial
    count = count + 1
    factorial = factorial * count

print("The approximation of e is {}".format(e))
```

Exercise 8

Proposed solution:

```
value = int(input("Specify the starting value of the sequence: "))
while value <= 0:
    print("The starting value of the series must be greater than 0")
    value = int(input("Specify the starting value of the sequence: "))

threshold = int(input("Specify a threshold: "))
while threshold <= 0:
    print("Threshold must be greater than 0")
    threshold = int(input("Specify a threshold: "))

while value < threshold:
    print(value, end=" ")

    aux = value
    accum = 1
    while aux > 0:
        if aux % 10 != 0:
            accum = accum * (aux % 10)
        aux = aux // 10

    value = value + accum
```



Exercise 9

Proposed solution:

```
value = int(input("Specify the starting value of the sequence: "))
while value <= 0:
    print("The starting value of the series must be greater than 0")
    value = int(input("Specify the starting value of the sequence: "))

n = int(input("Specify the number of elements to show: "))
while n <= 0:
    print("The number of elements to show must be greater than 0")
    n = int(input("Specify the number of elements to show: "))

for i in range(n):
    print(value, end=" ")

    aux = value
    accum = 1
    while aux > 0:
        if aux % 10 != 0:
            accum = accum * (aux % 10)
        aux = aux // 10

    value = value + accum
```

Exercise 10

Proposed solution:

```
x = int(input("Specify the number whose multiples will be shown: "))
a = int(input("Specify the beginning of the interval: "))
b = int(input("Specify the end of the interval: "))

if a % x != 0:
    a = a + x - a % x

for i in range(a, b + 1, x):
    print(i, end=" ")
```

Exercise 11

S_i	x	n	$n > 0$
S_0	0	19	True
S_1	1	9	True
S_2	2	4	True
S_3	3	2	True
S_4	4	1	True
S_5	5	0	False

The value of x after the execution of the program is 5.



Exercise 12

At first glance, the program is composed of two nested for loops. In this regard, it is worth calculating how many iterations each of them is going to be executed.

- After inspection of the range method call in the first-level for loop, it can be seen that it executes a total of 5 times, in such a way that the control variable `i` is assigned the values 0, 1, 2, 3 and 4 in each of the iterations.
- The second-level for loop, in turn, is executed `i + 1` iterations for each of iteration of the first-level for loop. In this way, it will be executed once the first time, so that the control variable `j` is assigned the value 0; it will be executed twice the second time, so that the control variable `j` is assigned the values 0 and 1, and so on for 5 iterations total.

Taking into account that the value of variable `j` is printed in each iteration of the second-level for loop, the output of the exercise would be:

```
0 0 1 0 1 2 0 1 2 3 0 1 2 3 4
```