COMP 204

Debugging skills and Nested loops

Mathieu Blanchette

Quiz 7 password

Testing your program

Once you've written a first draft of your program, you need to test it:

- Provide some input data, and verify manually that the output is correct
- Ask yourself: what kind of situation could break my program? Test them! Do not limit yourself to the examples provided in the assignments
- Be mean: Test the "boundary cases", i.e. the smallest and largest inputs that make sense in the context of the problem.
  - For the last question of assignment #1 (protein length):
  - What if the gene sequence has no start codon?
  - What if it has no in-frame stop codon?
  - What if it has an out-of-frame stop codon?

Software testing is not easy! It is actually an entire branch of computer science!

Live demo on Spyder

Task: Write a program that allows the user to enter numbers, one by one, until they type "done". Then, the program reports the average, minimum, and maximum of the values entered.

Assume that the user will only enter numbers or "done".

Nested loops

Just like nested conditionals, we can have nested loops.

```
1 while booleanExpression1:
2     # beginning of the outer loop
3     while booleanExpression2:
4         # body of the inner loop
5     # rest of the outer loop
6
7 # rest of program (outside while loop)
```

Execution:

- Line 1: booleanCondition1 is evaluated. If not true, jump to line 7. If true go to line 2
- Line 2: execute "beginning of outer loop"
- Line 3: booleanCondition2 is evaluated. If not true, jump to line 5. If true go to line 4
- Line 4: Execute body of inner loop
- After line 4: Return to line 3
- Line 5: execute rest of outer loop
- After line 5: Return to line 1
- Line 7: execute rest of program

Windchill

Background: The windchill index measures the sensation of cold on exposed skin. It is determined by the temperature (Celsius) and the wind speed (km/h). The formula is $windChill = 13.12 + 0.6215 * T - 11.37 * W^{0.16} + 0.2936 * T * W^{0.16}$, where T is the temperature and W is the wind speed.

Task: Repeatedly ask the user to enter the temperature (stop when the user enters "done"), and then ask for the minimum windchill the user can tolerate. Then, print out the *highest* windspeed for which the windchill index drops below the tolerable value.

Example: If the user enters a temperature of -10 and a tolerable windchill of -25, your program should report that the user can tolerate windspeeds of up to 98 km, because this is the point where the windchill drops below -25 when it is -10C.

Nested loops - Windchill calculator

```python
import math

entry=""
while entry!="done":
    entry=input("Enter temperature (C):")
    if entry!="done":
        temp = float(entry)
        tol_windchill = float(input("Enter \
                                     tolerable windchill:"))

        # Use while loop to look for wind speed that results
        # in an untolerable windchill
        wind=0
        while True:   # keep looping until we hit a break
            windchill = 13.12 + 0.6215*temp - \
                        11.37 * math.pow(wind,0.16) + \
                        0.3965 * temp * math.pow(wind,0.16)
            if windchill<tol_windchill:
                break  # we've reached a wind strong enough
            wind += 1

        print("You can tolerate a wind speed of up to:", \
              wind,"km/h")
```

Nested loops - Windchill calculator

```python
import math

entry=""
while entry!="done":
    entry=input("Enter temperature (C):")
    if entry!="done":
        temp = float(entry)
        tol_windchill = float(input("Enter \
                                    tolerable windchill:"))

        # Use while loop to look for wind speed that results
        # in an untolerable windchill
        wind=0
        while True:   # keep looping until we hit a break
            windchill = 13.12 + 0.6215*temp - \
                        11.37 * math.pow(wind,0.16) + \
                        0.3965 * temp * math.pow(wind,0.16)
            if windchill<tol_windchill:
                break  # we've reached a wind strong enough
            wind += 1

        print("You can tolerate a wind speed of up to:", \
              wind,"km/h")
```

Nested loops example 1 - BMI table

Print the BMI for every combination of weights and heights.
Weight should range from 50 kg to 70 kg (in increment of 10).
Height should range from 1.6 m to 1.8m, in increment of 0.1m.
Output should look like this:

```
BMI for 50 kg, 1.6 m is 19.53
BMI for 50 kg, 1.7 m is 17.30
BMI for 50 kg, 1.8 m is 15.42
BMI for 60 kg, 1.6 m is 23.43
...
BMI for 70 kg, 1.8m is 21.60
```

Algorithm:

- Use a loop to iterate through weights from 50 to 70 by 10
    - Use an inner loop to iterate through heights from 1.0 to 2.0
    - Calculate BMI from current values of weight and height, print

Nested loops - BMI table

```python
weight = 50
while weight <= 70:
    height = 1.6 # reset height to 1.6 INSIDE the loop
    while height < 1.9:
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
        height = height + 0.1
    weight = weight + 10
```

Nested loops - BMI table

```python
weight = 50
while weight <= 70:
    height = 1.6 # reset height to 1.6 INSIDE the loop
    while height < 1.9:
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
        height = height + 0.1
    weight = weight + 10
```

```python
# What's wrong with this code?
weight = 50
height = 1.6 # reset height to 1.6 OUTSIDE of the loop
while weight <= 80:
    while height < 1.9:
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
        height = height + 0.1
    weight = weight + 10
```

Nested loops - BMI table

```python
weight = 50
while weight <= 70:
    height = 1.6 # reset height to 1.6 INSIDE the loop
    while height < 1.9:
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
        height = height + 0.1
    weight = weight + 10
```

```python
# What's wrong with this code?
weight = 50
height = 1.6 # reset height to 1.6 OUTSIDE of the loop
while weight <= 80:
    while height < 1.9:
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
        height = height + 0.1
    weight = weight + 10
```

```python
import numpy as np # for floating-point range function
for weight in range(50,80,10): # for-loop
#     for height in np.arange(1.6,1.9,0.1): # for-loop
    for height in np.arange(1.6,1.9,0.1): # for-loop
        BMI = weight/(height**2)
        print("BMI for", weight," kg,", height," m is ",BMI)
```

Nested loops example 2 - Prime numbers

A prime number is a number that is divisible only by 1 and itself.
Task: Print all prime numbers up to a given limit.

Algorithm:

- Use a loop to enumerate each candidate number, starting from 2 up to the given number
  - Test each candidate by using a second loop that enumerates every possible factor of the candidate prime, from 2 up to squared root of the candidate number
  - If never found a factor, then the number is prime. Print it.

Nested loops - Prime numbers

```python
import math
maxNumber = int(input("Enter max. number to consider: "))

candidatePrime = 2
while candidatePrime <= maxNumber:

    isPrime = True  # By default the number is prime
    candidateFactor = 2  # Test at all possible factors
                         # of candidatePrime,starting with 2
    while candidateFactor <= math.sqrt(candidatePrime):
        # if the remainder of the integer division is zero,
        # then candidateFactor is a factor of candidatePrime,
        # so candidatePrime is not prime
        if candidatePrime % candidateFactor == 0:
            isPrime = False
            break;  # break out of the inner loop, since
                    # we've found a factor

        candidateFactor = candidateFactor + 1

    if isPrime:
        print(candidatePrime)

    candidatePrime = candidatePrime + 1
```

Nested loops - Prime numbers

```python
# for-loop version
import numpy as np
maxNumber = int(input("Enter max. number to consider: "))

candidatePrime = 2

for candidatePrime in range(2, maxNumber+1):

    isPrime = True  # By default the number is prime
    candidateFactor = 2  # Test at all possible factors
                         # of candidatePrime,starting with 2
    for candidateFactor in np.arange(2, np.sqrt(candidatePrime)):

        if candidatePrime % candidateFactor == 0:
            isPrime = False
            break;  # if not prime break out of the inner loop
    if isPrime:
        print(candidatePrime)
```