FIT1045/FIT1053 Algorithmic Problem Solving – Tutorial 2.

Objectives

After this tutorial, you should be able to:

- Interpret and use boolean expressions.

- Interpret and use conditional statements.

- Create an algorithmic solution to a simple problem.

- Walk through simple unknown code.

Prepared Question

Complete a code trace on the following function call by transcribing the function and noting the values of the important variables at the indicated points in the function.

```python
def accept_hypothesis(mu_hat, mu, sigma, n):
    z_score = (mu_hat - mu) / (sigma/n**0.5)
    # z_score = ?
    if z_score < 0:
        z_score *= -1
    # z_score = ?
    P = 0
    if z_score < 0.5:
        P = 0.5
    elif z_score < 1:
        P = 0.3
    elif z_score < 2:
        P = 0.15
    else:
        P = 0.02
    # P = ?
    p_value = 2*P
    # p_value = ?
    return p_value <= 0.05

answer = accept_hypothesis(5, 20, 4, 16)
# answer = ?
```

Workbook

§1.1 Numerical Expressions, §1.2 Boolean Expressions, and §2.1 Conditionals.
While your tutors mark the prepared question, attempt the above Workbook sections. Your tutors will go through some of these questions with the class.

Warm-up

Do not solve these questions by running in Python.

Numerical expressions: What does each expression yield?

- `(2+7)/(5%3)`

- `5**9**(1/2) // (3*2)`

Boolean expressions: What does each expression yield?

- `0 or None and False`

- `2<2 and 'cat' or 7 > 4 < 12 or not []`

Conditionals: What is returned by each function call?

- ```
def is_prime(x):
    response = 'unknown'
    if x%2 == 0:
        response = 'is even'
    elif x%3 == 0 or x%5 == 0 or x%7 == 0:
        response = 'is composite'
    elif x < 13:
        response = 'is prime'
    else:
        response = 'is too big'
    return response

  is_prime(15)
  ```

- ```
def is_date(date, month):
    if (month==4 or month==6 or
        month==9 or month==11):
        return 0<date<=30
    if month==2:
        return date<=28
    return (0<date<=31 and 1<=month<=12)

is_date(31, 9)
is_date(8, 7)
  ```

Code trace: What is returned by each function call?

- ```
def mystery1(word, letter):
    i = 0
    while i < len(word):
        if word[i] == letter:
            return True
        i += 1
    return False

  mystery('anyway', 'e')
  ```

- ```
def mystery2(word, letter):
    i = 0
    count = 0
    while i < len(word):
        if word[i] == letter:
            count += 1
        i += 1
    return count

mystery('anyway', 'a')
  ```

Real Roots

1. Write an algorithm in English for finding the real roots[1] of a quadratic equation $ax^2 + bx + c = 0$ for arbitrary real coefficients a, b and c. Things to note:

   (a) The quadratic formula is $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

   (b) Begin by considering the inputs and outputs of this algorithm.

   (c) There exist 0, 1, or 2 real roots in any quadratic equation, depending on what is yielded by $b^2 - 4ac$.

   (d) The number of real roots affects how the quadratic formula should be used.

   (e) Determining how many real roots you are trying to find should occur somewhere in the algorithm.

2. Structure your algorithm as a function `real_roots(a, b, c)` which takes as input the three coefficients of a polynomial and returns a list of the roots. If there are no roots, the list should be empty.

3. Implement your algorithm in Python.

Extension Questions

1. Describe the standard algorithm for finding the decimal representation of a positive binary number. (For example, how to convert 100101 to 37. If you are unsure how to convert a binary number to its decimal representation, ask your tutor for guidance.) Write the algorithm in English, and then implement your algorithm in Python.

2. Describe in English an algorithm for taking an unsorted sequence of numbers and sorting them into ascending order. Be as specific as possible.

Unscrambling Code

In this task you will try to unscramble lines of code to create a function with the desired behaviour. Feel free to either write each line onto a sheet of paper and cut it into separate lines, or to simply write down the unscrambled function directly. Remember to ensure your indentation is correct.

Extension: Some of these implementations are unnecessarily verbose. Try to make them more succinct.

Part A - Magic Number

Arrange the lines such that you produce a function that takes three integers, `x`, `y`, and `n`, that returns `True` if one of `x` or `y` is `n`, or if `x` and `y` add to `n`, or if their difference is `n`.

- return True
- return False
- return True
- else:
- def magic_num(x, y, n):
- return True
- elif x+y == n:
- if x == n or y == n:
- elif x-y == n or y-x == n:

---

[1] the x-intercepts

Part B - Icecream

Arrange the lines such that you produce a function that takes an integer, x, and a boolean, summer, that returns the number of icecreams you need to buy. The input x tells you how many icecreams you already have, and summer tells you whether it is summer. You must always have at least 2 icecreams, except during summer when you must have 10. If you have enough or more than enough icecreams, you do not need to buy any.

- return 0

- if 10-x > 0:

- return 2-x

- def icecreams(x, summer):

- return 10-x

- elif 2-x > 0:

- if summer:

Part C - Rounding

Arrange the lines such that you produce a function that takes an integer n and returns that number rounded to its nearest 10. If the number in the ones column is greater than 5, round up; if the number in the ones column is less than 5, round down. If the number in the ones column is 5, round in such a way that the number in the tens column is even.

- num = n-remainder

- elif remainder > 5:

- if remainder < 5:

- if num//10%2 == 0:

- return num

- else:

- return num

- remainder = n%10

- return num + 10

- else:

- return num + 10

- def rounding(n):

Puzzles - FIT1053

Use these puzzles to start practicing open-ended problem solving. None of these are examinable, and solutions will not be provided.

1. Over a 24 hour period, how many times will the hour hand and the minute hand overlap on an analogue clock?

2. 99 unique numbers between 1 and 100 are listed one by one, with five second pause after every number. If you are not allowed to take any notes, what is the best way to determine which is the missing number?

3. You have a drawer with 10 pairs of black socks and 10 pairs of white socks. How many times do you need to grab a sock, to be sure that you have a matching pair?

4. Is the number 9991 prime?

5. There are n closed doors in a hallway, numbered sequentially from 1 to n. You walk down the corridor n times, and at the end of the corridor you run back to the start. On your ith trip down the hall you toggle every ith door (open it if it's closed, close if it's opened). After the last trip down the corridor, which doors are opened, and which are closed?

6. You are trying to get a wolf, a goat, and a cabbage across a river. You have a boat that can only hold two entities at once. If you leave the wolf alone with the goat, the wolf will eat the goat. If you leave the goat alone with the cabbage, the goat will eat the cabbage. What do you do?

7. Mario asks a tutor to conduct a poll at their consultation for which day of the week the final exam should be held on. The tutor reports back the following: 73% say Monday, 58% say Wednesday, and 32% say Friday. The tutor admits they were a little lazy in their reporting, and simply dropped all decimal values (i.e. both 5.01% and 5.99% would have been reported as 5%). Mario realises two things: firstly, some students must have voted more than once; and secondly, he can now see whether the consult was well attended. What was the minimum number of students at the consult?