COMP 1023 Introduction to Python Programming

Exercises on Branching (a.k.a. Checking & Selection)

COMP 1023 Teaching Team

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China

Exercise 1

The two roots of a quadratic equation, for example $ax^2 + bx + c = 0$, can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$b^2 - 4ac$ is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Write a program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display "The equation has no real roots".

```
Enter a: 13.5
Enter b: 45.2
Enter c: 12.4
The roots are -0.3014932803673518 and -3.046664867780797
```

Exercise 1 - Solutions

```python
import math

# Prompt user for coefficients
a = float(input("Enter a: "))
b = float(input("Enter b: "))
c = float(input("Enter c: "))

# Calculate discriminant
discriminant = b**2 - 4*a*c

# Check the discriminant and calculate roots accordingly
if discriminant > 0:
    # Two real roots
    root1 = (-b + math.sqrt(discriminant)) / (2*a)
    root2 = (-b - math.sqrt(discriminant)) / (2*a)
    print(f"The roots are {root1} and {root2}")
elif discriminant == 0:
    # One real root
    root = -b / (2*a)
    print(f"The root is {root}")
else:
    # No real roots
    print("The equation has no real roots")
```

Exercise 2

Revise the following program to generate one single-digits integer and prompt the user to enter the square of the integer.

```
import random
number1 = random.randint(0, 9)
number2 = random.randint(0, 9)
answer = int(input("What is", number1, "+", number2, "? "))
print(number1, "+", number2, "=", answer, "is", number1 + number2 == answer)
```

```
What is square of 4? 16
Square of 4 = 16 is True

What is square of 6? 23
Square of 6 = 23 is False
```

Exercise 2 - Solutions

```
import random

number = random.randint(0, 9)
answer = int(input(f"What is square of {number}? "))

print(f"Square of {number} = {answer} is", number * number == answer)
```

Exercise 3

Solve the following 2×2 system of linear equation:

$$ax + by = e$$
$$cx + dy = f$$
$$x = \frac{ed - bf}{ad - bc}$$
$$y = \frac{af - ec}{ad - bc}$$

Write a program that prompts the user to enter a, b, c, d, e, ad f and display the result. If ad - bc is 0, report that "The equation has no solution".

```
Enter a: 9.0
Enter b: 4.0
Enter c: 3.0
Enter d: -5.0
Enter e: -6.0
Enter f: -21.0
x is -2.0 and y is 3.0
```

Exercise 3 - Solutions

```python
# Prompt user for coefficients
a = float(input("Enter a: "))
b = float(input("Enter b: "))
c = float(input("Enter c: "))
d = float(input("Enter d: "))
e = float(input("Enter e: "))
f = float(input("Enter f: "))

# Calculate the determinant
determinant = a * d - b * c

# Check if the system has a solution
if determinant == 0:
    print("The equation has no solution")
else:
    # Calculate x and y using Cramer's rule
    x = (e * d - b * f) / determinant
    y = (a * f - e * c) / determinant
    print(f"x is {x} and y is {y}")
```

Exercise 4

Write a program that prompts the user to enter an integer for the current month (January is 0, February is 1, ..., and December is 11). Also, prompt the user to enter the number of months elapsed from the current month and display the future month of the year.

```
Enter current month: 4
Enter the number of months elapsed since current month: 6
Current month is May and the future month is November
```

Exercise 4 - Solutions

```python
# List of month names
months = [
    "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
]

# Prompt user for inputs
current_month = int(input("Enter current month: "))
elapsed_months = int(input("Enter the number of months elapsed since current month: "))

# Calculate future month using modulo 12 to wrap around the year
future_month = (current_month + elapsed_months) % 12

# Display results
print(f"Current month is {months[current_month]} and the future month is {months[future_month]}")
```

Exercise 5

Suppose you are planning to commute with an option of two trains. Write a program to find the train with fastest route that takes minimum time. The program prompts the user to enter the route length and average speed of each train and then displays the one with the fastest route.

```
Enter length (kms) of the route 1: 100
Enter speed (km/h) of the train 1: 80
Enter length (kms) of the route 2: 120
Enter speed (km/h) of the train 2: 100
Train 2 has the fastest route.
```

Exercise 5 - Solutions

```python
# Prompt user for train 1 details
route1_length = float(input("Enter length (kms) of the route 1: "))
train1_speed = float(input("Enter speed (km/h) of the train 1: "))

# Prompt user for train 2 details
route2_length = float(input("Enter length (kms) of the route 2: "))
train2_speed = float(input("Enter speed (km/h) of the train 2: "))

# Calculate travel time for each train (time = distance / speed)
time1 = route1_length / train1_speed
time2 = route2_length / train2_speed

# Compare times and display the result
if time1 < time2:
    print("Train 1 has the fastest route.")
elif time2 < time1:
    print("Train 2 has the fastest route.")
else:
    print("Both trains have the same travel time.")
```

Exercise 6

Randomly generates a multiplication question with two integers less than 100. Write a program to prompt the user to enter the product of two numbers and then displays whether the answer is correct or wrong, and show the correct answer.

```
What is 86 * 28? 27
Your answer is wrong.
86 * 28 should be 2408
```

Exercise 6 - Solutions

```python
import random

# Generate two random integers less than 100
number1 = random.randint(0, 99)
number2 = random.randint(0, 99)

# Calculate the correct product
correct_answer = number1 * number2

# Prompt user for answer
user_answer = int(input(f"What is {number1} * {number2}? "))

# Check if the answer is correct and display result
if user_answer == correct_answer:
    print("Your answer is correct.")
else:
    print("Your answer is wrong.")
    print(f"{number1} * {number2} should be {correct_answer}")
```

Exercise 7

Write a program that prompts the user to enter an integer and checks whether the number is divisible by both 2 and 3, divisible by 2 or 3, or just one of them (but not both).

```
Enter an integer: 30
Is 30 divisible by 2 and 3? True
Is 30 divisible by 2 or 3? True
Is 30 divisible by 2 or 3, but not both? False
```

Exercise 7 - Solutions

```python
# Prompt user for an integer
number = int(input("Enter an integer: "))

# Check divisibility
divisible_by_2 = number % 2 == 0
divisible_by_3 = number % 3 == 0

# Calculate the three conditions
both_2_and_3 = divisible_by_2 and divisible_by_3
either_2_or_3 = divisible_by_2 or divisible_by_3
exactly_one = (divisible_by_2 or divisible_by_3) and not both_2_and_3

# Display results
print(f"Is {number} divisible by 2 and 3? {both_2_and_3}")
print(f"Is {number} divisible by 2 or 3? {either_2_or_3}")
print(f"Is {number} divisible by 2 or 3, but not both? {exactly_one}")
```

Exercise 8

Write a program that lets the user guess whether a flipped coin displays the head or the tail. The program randomly generates an integer 0 or 1, which represents head or tail. The program prompts the user to enter a guess and reports whether the guess is correct or incorrect.

```
Guess head or tail?
Enter 0 for head and 1 for tail: 0
Sorry, it is a tail.
```

Exercise 8 - Solutions

```python
import random

# Generate random coin flip (0 for head, 1 for tail)
coin = random.randint(0, 1)

# Prompt user for guess
print("Guess head or tail?")
guess = int(input("Enter 0 for head and 1 for tail: "))

# Check if guess is correct and display result
if guess == coin:
    if coin == 0:
        print("Congratulations! It is a head.")
    else:
        print("Congratulations! It is a tail.")
else:
    if coin == 0:
        print("Sorry, it is a head.")
    else:
        print("Sorry, it is a tail.")
```

Exercise 9

Write a program that prompts the user to enter a four-digit integer and displays the number in reverse order.

```
Enter an integer: 3125
The reversed number is 5213
```

Exercise 9 - Solutions

```python
# Prompt user for a four-digit integer
number = int(input("Enter an integer: "))

# Extract each digit using integer division and modulo
digit1 = number // 1000
digit2 = (number // 100) % 10
digit3 = (number // 10) % 10
digit4 = number % 10

# Reverse the digits
reversed_number = digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1

# Display the reversed number
print(f"The reversed number is {reversed_number}")
```

Exercise 10

Write a program that plays the popular scissor-rock-paper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number 0, 1, or 2 representing scissor, rock, and paper. The program prompts the user to enter a number 0, 1, or 2 and displays a message indicating whether the user or the computer wins, loses, or draws.

```
Scissor (0), rock (1), paper (2): 2
The computer is paper. You are paper too. It is a draw.
```

Exercise 10 - Solutions

```python
import random

# Generate computer's choice (0: scissor, 1: rock, 2: paper)
computer = random.randint(0, 2)

# Prompt user for choice
user = int(input("Scissor (0), rock (1), paper (2): "))

# Define choices for display
choices = ["scissor", "rock", "paper"]

# Display choices
print(f"The computer is {choices[computer]}. You are {choices[user]}.")

# Determine the winner
if computer == user:
    print("It is a draw.")
elif (user == 0 and computer == 2) or (user == 1 and computer == 0) or (user == 2 and computer == 1):
    print("You win!")
else:
    print("You lose!")
```

Exercise 11

Write a program that reads four integer numbers and checks whether the numbers are all positive and entered in ascending order or not. The program displays a message indicating whether the numbers are in ascending order or not based on the order of entered numbers.

```
Enter number 1: 2
Enter number 2: 4
Enter number 3: 6
Enter number 4: 9
The numbers are positive and entered in ascending order
(2, 4, 6, 9)
```

Exercise 11 - Solutions

```python
# Read four integer numbers
num1 = int(input("Enter number 1: "))
num2 = int(input("Enter number 2: "))
num3 = int(input("Enter number 3: "))
num4 = int(input("Enter number 4: "))

# Check if all numbers are positive and in ascending order
all_positive = num1 > 0 and num2 > 0 and num3 > 0 and num4 > 0
in_ascending_order = num1 < num2 < num3 < num4

# Display results
if all_positive and in_ascending_order:
    print("The numbers are positive and entered in ascending order")
    print(f"({num1}, {num2}, {num3}, {num4})")
else:
    print("The numbers are not both positive and in ascending order")
```

Exercise 12

Write a program that prompts the user to enter a point (x, y) and checks whether this point is within the circle centered at (0, 0) with radius 10. For example, (4, 5) is inside the circle and (9, 9) is outside the circle.

(Hint: A point is in the circle if its distance (0, 0) is less than or equal to 10. The formula for computing the distance is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Test your program to cover all cases.

```
Enter the x-coordinate of the point: 4.5
Enter the y-coordinate of the point: 6.8
Point (4.5, 6.8) is in the circle
```

Exercise 12 - Solutions

```python
import math

# Prompt user for point coordinates
x = float(input("Enter the x-coordinate of the point: "))
y = float(input("Enter the y-coordinate of the point: "))

# Calculate distance from origin (0, 0)
distance = math.sqrt(x**2 + y**2)

# Check if point is within the circle
if distance <= 10:
    print(f"Point ({x}, {y}) is in the circle")
else:
    print(f"Point ({x}, {y}) is not in the circle")
```

Exercise 13

Two points on line 1 are given as (x1, y1) and (x2, y2) and one line 2 as (x3, y3) and (x4, y4). The intersecting point of the two lines can be found by solving the following linear equation:

$$(y_1 - y_2)x - (x_1 - x_2)y = (y_1 - y_2)x_1 - (x_1 - x_2)y_1$$
$$(y_3 - y_4)x - (x_3 - x_4)y = (y_3 - y_4)x_3 - (x_3 - x_4)y_3$$

If the equation has o solutions, the two lines are parallel. Write a program that prompts the user to enter four points and displays the intersecting point.

```
Enter x1: 2.4
Enter y1: 5.6
Enter x2: 7.3
Enter y2: 2.1
Enter x3: -4.5
Enter y3: 4.5
Enter x4: -3.4
Enter y4: -9.2
The intersecting point is at (-5.013495575221239, 10.895353982300886)
```

Exercise 13 - Solutions

```
# Prompt user for four points
x1 = float(input("Enter x1: ")); y1 = float(input("Enter y1: "))
x2 = float(input("Enter x2: ")); y2 = float(input("Enter y2: "))
x3 = float(input("Enter x3: ")); y3 = float(input("Enter y3: "))
x4 = float(input("Enter x4: ")); y4 = float(input("Enter y4: "))

# Calculate coefficients for the linear equations
# Equation 1: (y1 - y2)x - (x1 - x2)y = (y1 - y2)x1 - (x1 - x2)y1
a1 = y1 - y2; b1 = -(x1 - x2); c1 = (y1 - y2) * x1 - (x1 - x2) * y1
# Equation 2: (y3 - y4)x - (x3 - x4)y = (y3 - y4)x3 - (x3 - x4)y3
a2 = y3 - y4; b2 = -(x3 - x4); c2 = (y3 - y4) * x3 - (x3 - x4) * y3

# Calculate the determinant
determinant = a1 * b2 - a2 * b1

# Check if lines are parallel
if determinant == 0:
    print("The two lines are parallel and have no intersection point.")
else:
    # Calculate the intersection point using Cramer's rule
    x = (c1 * b2 - c2 * b1) / determinant
    y = (a1 * c2 - a2 * c1) / determinant
    print(f"The intersecting point is at ({x}, {y})")
```

Exercise 14

Write a program that prompts the user to enter a 5-digit integer and determines whether it is a palindrome number. A number if palindrome if it reads the same from right to left and from left to right.

```
Enter an integer: 15651
15651 is a palindrome

Enter an integer: 24521
24521 is not a palindrome
```

Exercise 14 - Solutions

```python
# Prompt user for a 5-digit integer
number = int(input("Enter an integer: "))

# Extract each digit
digit1 = number // 10000
digit2 = (number // 1000) % 10
digit3 = (number // 100) % 10
digit4 = (number // 10) % 10
digit5 = number % 10

# Check if it's a palindrome
if digit1 == digit5 and digit2 == digit4:
    print(f"{number} is a palindrome")
else:
    print(f"{number} is not a palindrome")
```

Exercise 15

Write a program that prompts the user to enter the center x-, y-coordinates, width, and height of two rectangles and determines whether the second rectangle is inside the first or overlaps with the first. Test your program to cover all cases.

```
Enter r1's center x-coord: 2.5
Enter r1's center y-coord: 4.5
Enter r1's width: 3.7
Enter r1's height: 9.2
Enter r2's center x-coord: 1.5
Enter r2's center y-coord: 4.5
Enter r2's width: 0.7
Enter r2's height: -9.2
r2 is inside r1
```

Exercise 15 - Solutions

```python
# Get rectangle details
r1_x, r1_y = float(input("Enter r1's center x: ")), float(input("Enter r1's center y: "))
r1_w, r1_h = abs(float(input("Enter r1's width: "))), abs(float(input("Enter r1's height: ")))
r2_x, r2_y = float(input("Enter r2's center x: ")), float(input("Enter r2's center y: "))
r2_w, r2_h = abs(float(input("Enter r2's width: "))), abs(float(input("Enter r2's height: ")))

# Calculate boundaries
r1_l, r1_r = r1_x - r1_w/2, r1_x + r1_w/2
r1_b, r1_t = r1_y - r1_h/2, r1_y + r1_h/2
r2_l, r2_r = r2_x - r2_w/2, r2_x + r2_w/2
r2_b, r2_t = r2_y - r2_h/2, r2_y + r2_h/2

# Check relationship
if r2_l >= r1_l and r2_r <= r1_r and r2_b >= r1_b and r2_t <= r1_t:
    print("r2 is inside r1")
elif r2_l <= r1_r and r2_r >= r1_l and r2_b <= r1_t and r2_t >= r1_b:
    print("r2 overlaps r1")
else:
    print("r2 does not overlap r1")
```

Exercise 16

An ISBN-10 (International Standard Book Number) consists of 10 digits: $d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10}$. The last digit, d_{10}, is a check-sum, which is calculated from the other nine digits using the following formula:

$$(d_1 \times x1 + d_2 \times 2 + d_3 \times 3 + d_4 \times 4 + d_5 \times 5 + d_6 \times 6 + d_7 \times 7 + d_8 \times 8 + d_9 \times 9)\%11$$

If the checksum is 10, the last digit is denoted as X according to the ISBN-10 convention. Write a program that prompts the user to enter the first 9 digits and displays the checksum. Your program should read the input as an integer. If the integer starts with 0s, don't enter these zeros in the input.

```
Enter the first 9 digits of an ISBN an integer: 123601267
The checksum is 4
```

Exercise 16 - Solutions

```python
# Prompt user for first 9 digits
isbn = input("Enter the first 9 digits of an ISBN as an integer: ")

# Calculate checksum without loop
checksum = (int(isbn[0]) * 1 + int(isbn[1]) * 2 + int(isbn[2]) * 3 +
            int(isbn[3]) * 4 + int(isbn[4]) * 5 + int(isbn[5]) * 6 +
            int(isbn[6]) * 7 + int(isbn[7]) * 8 + int(isbn[8]) * 9) % 11

# Display result
print(f"The checksum is {'X' if checksum == 10 else checksum}")
```

Exercise 17

Write a program that displays a random coordinate inside a square. The square is centered at (0, 0) and the length of each side is 100.

```
35 21
```

Exercise 17 - Solutions

```
import random

x = random.randint(-50, 50)
y = random.randint(-50, 50)
print(x, y)
```