Balchunas

AP CSP: Problem Set 0

Specification

Submit a .py file named `your_name_pset0.py`. Your file should **NOT** execute any functions. You are just filling them in. When I run your program there should be no errors and no output. You may choose to test your functions as you go, but you may delete the function calls after you test or comment them out. Comment your problems with the problem number above the function header.

Problem 1

The area of a rectangle is given as the length of the rectangle multiplied by the width of the rectangle. Given the length and width, return the area of the rectangle.

```
rect_area(length, width)
rect_area(2, 3) # 6
rect_area(3, 3) # 9
```

Problem 2

The area of a circle is given as πr^2 where r is the radius of the circle. Given the radius of a circle, return the surface area of the circle. Use 3.14 as π.

```
circ_area(radius)
circ_area(2) # 12.56
circ_area(4) # 50.24
```

Problem 3

The surface area of a cylinder is given as the area of the circular top and bottom bases, plus the lateral surface area of the curved outside. The equation is $2\pi rh + 2\pi r^2$ where r is the radius of the cylinder, and h is the height of the cylinder. Given a cylinder's radiusand height, return the surface area of a cylinder. Use 3.14 as π.

```
cylinder_area(radius, height)
cylinder_area(2, 4)  # 75.36
cylinder_area(4, 10) # 351.68
```

Problem 4

The surface area of a sphere is given as $4\pi r^2$ where r is the radius of the sphere. Given the radius of a circle, return the surface area of the sphere. Use 3.14 as π.

```
sphere_surface_area(radius)
sphere_surface_area(2) # 50.24
sphere_surface_area(4) # 200.96
```

Problem 5

Given two integers `num1` and `num2`, return their sum. Unless the two values are the same, then return double their sum.

```
sum_double(a, b)
sum_double(5, 5) # 20
sum_double(1, 3) # 4
```

Problem 6

Given 2 integers `num1` and `num2`, return true if they are both in the range 20 to 30 inclusive, or they are both in the range 90 to 100 inclusive.

```
in_range(num1, num2)
in_range(20, 31) # False
in_range(20, 21) # True
in_range(90, 100)# True
```

Problem 7

The euclidean distance between any set of points (x_1, y_1) and (x_2, y_2) is given by $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Write a function that calculates the distance between the origin $(0, 0)$ and a point (x, y).

```
distance(x, y)
distance(2, 2) # 2.828
distance(1, 5) # 5.099
```

Problem 8

A motor controlling a gate should activate if it is a workday, if the gate is activated by someone driving up, and if the gate is down. Given Boolean variables: `workday` which is `True` if it is a workday, `gate_activated` which is `True` if a car activates the gate, and `gate_up` which is `True` if the gate is up, return `True` if the motor should activate and `False` if the motor should not activate.

```
open_gate(workday, gate_activated, gate_up)
open_gate(True, True, True)  # False
open_gate(True, False, True) # False
open_gate(True, True, False) # True
```

Problem 9

A person over 24 years of age (inclusive) that has not had any accidents in the past 5 years is eligible for a discount on a rental car. Given a person's age and whether they have had any accidents in the past 5 years, return `True` if the person is eligible for a discount, and `False` otherwise.

```
has_discount(age, had_accident)
has_discount(22, False) # False
has_discount(26, True)  # False
has_discount(28, False) # True
```

Problem 10

A person should apply to a certain school if their GPA is equal to or over 3.9 and their ACT was over, but not equal to, 31. However, if their GPA is equal to or over 4.25 or their ACT is 34 or above, the person should apply anyway. Given a GPA, and ACT score, return `True` if the person should apply to the school, and `False` otherwise.

```
should_apply(gpa, act_score)
should_apply(3.9, 31) # False
should_apply(3.9, 32) # True
should_apply(3.6, 35) # True
should_apply(4.25, 28)# True
```

Problem 11

On starting your car, the computer throws a "change oil" notification if you have not changed your oil in 10,000 miles, or if you have not changed your oil in 365 days (exclusive). Given the number of miles since your last oil change, and the number of days since your last oil change, return `True` if the light should turn on, and `False` otherwise.

```
oil_light(miles, days)
oil_light(15000, 100) # True
oil_light(7500, 200)  # False
```

Problem 12

The absolute value function $|x|$ returns the non-negative value of x regardless of its sign. Write a function that performs this mathematical operation and returns the result. Do **NOT** use the built-in `abs` function that comes with the Python standard library. The goal of this problem is to write your own.

```
abs_val(x)
abs_val(-3) # 3
abs_val(3)  # 3
```

Problem 13

Given two values `a` and `b`, return the value which is closest to the nearest multiple of 10. If they are the same distance, return `a`. For example, given the numbers 2 and 79, the program would return 79 because it is closer to 80 than 2 is to 0. Hint: Use modulo.

```
closest_tens(a, b)
closest_tens(2, 79)  # 79
closest_tens(-2, 84) # -2
closest_tens(-2, 2)  # -2
```

Problem 14

The price of a flight is calculated using an algorithm involving the free space on the flight and the number of available seats. Flights that are outside of the three day window will determine their price only on the number of seats available. If a flight is more than 75% full, there is a 25% increase in ticket price. Flights within three days of departure will be given a 15% discount if they are less than 50% full or a 40% upcharge if they are more than 90% full. Note that this 40% upcharge is applied instead of the 25% increase, so you need not worry about both upcharges due to occupancy. Given the ticket price of a flight, whether or not the flight is happening in the next three days, and the percent of open seats on the plane, return the price of a ticket.

```
get_ticket_price(price, flying_soon, open_seat_pct)
get_ticket_price(100, False, 0.15) # 125
get_ticket_price(100, True, 0.15)  # 165
get_ticket_price(100, False, 0.1)  # 100
get_ticket_price(100, True, 0.95)  # 85
```

Problem 15

Given a number, return `True` if the number is even and the last digit is a 2, 4, or 8. Return `False` otherwise.

```
is_special(num)
is_special(200) # False
is_special(44)  # True
```

Problem 16

Given a 3 digit number, return the number reversed. The input must be between 100 and 999 inclusive.

```
rev_num(n)
rev_num(123) # 321
rev_num(100) # 1
rev_num(987) # 789
rev_num(111) # 111
```

Problem 17

A year is a leap year (a year with 366 days instead of 365 days) if the year is exactly divisible by 4, except for the years that are exactly divisible by 100. However, if a year is exactly divisible by 100 and 400 it is a leap year. Given a year $year$, return `True` if the year is a leap year, and `False` if it is not a leap year.

```
is_leap_year(year)
is_leap_year(1900) # False
is_leap_year(2004) # True
is_leap_year(2008) # True
is_leap_year(1999) # False
```

Problem 18

There is an algorithm to tell what day of the week January 1st falls on for a given year. The algorithm is as follows: Calculate two quantities: the last two digits of the year minus 1, `Y`, and the first two digits of the year `C`. Start with the number 29 and subtract 2 times `C`, then add `Y`. Perform integer division with `Y` by 4 and add that to the original quantity. Perform integer division with `C` by 4 and add that to the original quantity. Take this whole quantity and find it's remainder when dividing by 7. This number corresponds to a day of the week with 0 being Sunday, 1 being Monday, 2 being Tuesday, etc. Given a year, return the day of the week the year began.

```
first_day(year)
first_day(1978) # Sunday
first_day(2001) # Monday
first_day(1999) # Friday
first_day(1980) # Tuesday
```

Problem 19

Given a month, day, and year, determine the day of the year (1 to 366) the date is. The month is given as a number between 1 and 12, days between 1 and 31 and year in YYYY format.

```
day_of_year(month, day, year)
day_of_year(1, 1, 2000)   # 1
day_of_year(6, 10, 1999)  # 161
day_of_year(5, 6, 2000)   # 127
day_of_year(12, 31, 2000) # 366
```

Problem 20

Using the functions you created in the past questions, determine the day of the week for any day of the year given the month, day, and year. Again, note that 0 corresponds to Sunday, 1 to Monday, etc.

```
day_of_week(month, day, year)
day_of_week(1, 1, 2000)   # Saturday
day_of_week(6, 10, 1996)  # Monday
day_of_week(5, 6, 1989)   # Saturday
day_of_week(12, 31, 2000) # Sunday
```