

Karthik Nair | 3EA | 00229802021

# Functions

---

1. Write a function `cubesum()` that accepts an integer and returns the sum of the cubes of individual digits of that number. Use this function to make functions `PrintArmstrong()` and `isArmstrong()` to print Armstrong numbers and to find whether is an Armstrong number.

```
def cubesum(n):
    sum = 0
    while n>0:
        sum+=(n%10)**3
        n=n//10
    return sum

def isArmstrong(n):
    return n if n==cubesum(n) else False

def PrintArmstrong(n):
    if isArmstrong(n):
        print(f"{n} is an Armstrong number")
    else:
        print(f"{n} is not an Armstrong number")

if __name__ == "__main__":
    PrintArmstrong(153)
    PrintArmstrong(154)
```

2. Write a function `prodDigits()` that inputs a number and returns the product of digits of that number.

```
def prodDigits(n):
    prod = 1
    while n>0:
        prod*=(n%10)
        n=n//10
    return prod

if __name__ == "__main__":
    print(prodDigits(1234))
```

3. Write a function `sumPdivisors()` that finds the sum of proper divisors of a number.

Proper divisors of a number are those numbers by which the number is divisible, except the number itself.

For example proper divisors of 36 are 1, 2, 3, 4, 6, 9, 18.

```
def sumPdivisors(n):  
    sum = 0  
    for i in range(1,n):  
        if n%i==0:  
            print(i)  
            sum+=i  
    return sum  
  
if __name__ == "__main__":  
    print(sumPdivisors(36))
```

4. Let's use functions to calculate your trip's costs:

a. Define a function called `hotel_cost` with one argument `nights` as input. The hotel costs \$140 per night. So, the function `hotel_cost` should return `140 * nights`.

b. Define a function called `plane_ride_cost` that takes a string, `city`, as input. The function should return a different price depending on the location. Below are the valid destinations and their corresponding round-trip prices.

"Charlotte": 183

"Tampa": 220

"Pittsburgh": 222

"Los Angeles": 475

c. Below your existing code, define a function called `rental_car_cost` with an argument called `days`. Calculate the cost of renting the car: Every day you rent the car costs \$40. (`cost=40*days`) if you rent the car for 7 or more days, you get \$50 off your total (`cost-=50`). Alternatively (`elif`), if you rent the car for 3 or more days, you get \$20 off your total. You cannot get both of the above discounts. Return that cost.

d. Then, define a function called `trip_cost` that takes two arguments, `city` and `days`. Have your function return the sum of calling the `rental_car_cost(days)`, `hotel_cost(days)`, and `plane_ride_cost(city)` functions.

e. Modify your `trip_cost` function definition. Add a third argument, `spending_money`. Modify what the `trip_cost` function does. Add the variable `spending_money` to the sum that it returns.

```
def hotel_cost(nights):
    return 140 * nights

def plane_ride_cost(city):
    if city == "Charlotte":
        return 183
    elif city == "Tampa":
        return 220
    elif city == "Pittsburgh":
        return 222
    elif city == "Los Angeles":
        return 475
    else:
        return "Invalid destination"

def rental_car_cost(days):
    cost = 40 * days
    if days >= 7:
        cost -= 50
    elif days >= 3:
        cost -= 20
    return cost

def trip_cost(city, days, spending_money):
    return rental_car_cost(days) + hotel_cost(days) + plane_ride_cost(city) +
    spending_money

print(trip_cost("Los Angeles", 5, 1000))
```

5. Write a function that takes in a list of integers and returns True if it contains 007 in order

Sample function:

spy\_game([1,2,4,0,0,7,5]) --> True

spy\_game([1,0,2,4,0,5,7]) --> True

spy\_game([1,7,2,0,4,5,0]) --> False

```
def spy_game(nums):
    code = [0,0,7,'x']
    for num in nums:
        if num == code[0]:
            code.pop(0)
    return len(code) == 1
```

```
print(spy_game([1,2,4,0,0,7,5]))  
print(spy_game([1,0,2,4,0,5,7]))  
print(spy_game([1,7,2,0,4,5,0]))
```