

CS153/453 Fall 2017

HW 6

Due: Monday Oct 2, 2017 11:59pm (before midnight)

Task 1

Re-do Homework 2 Tasks 1 and 2 with a function. The function should take in a parameter for **seconds**, and return the equivalent time in **years**, **days**, **hours**, **minutes**, and **seconds** using a tuple. In the main program, convert one trillion seconds into the equivalent time in years, days, hours, minutes and seconds, and print the result.

Task 2

Re-do Homework 2 Tasks 3 and 4 with a function. The function should take in parameters for **years**, **days**, **hours**, **minutes**, and **seconds**, and return the equivalent time in **seconds**. In the main program, convert 50 years 300 days 20 hours 50 minutes and 20 seconds into the equivalent time in seconds, print the result.

Task 3

Re-do Homework 1 using a function.

```
def barsRedeemed( coupons )
    return # codes to be completed

def bars( money )
    return money + barsRedeemed( money )

# main program
money = int(input("Enter an amount of money in dollars"))
print("For %d dollars, one can have %d bars" % (money, bars(money)))
```

Complete the codes for the function **barsRedeemed**. You can complete the codes using a **return** statement. The logic is

```
return BASE-VALUE if BASE-CASE-CONDITION else GENERAL-CASE
```

You need to rewrite the parts for **BASE-VALUE**, **BASE-CASE-CONDITION** and **GENERAL-CASE**.

Test your program with different test cases. Besides submitting the program, attach also screen shots for different test cases to your answers.

Task 4

Re-do Homework 2 Part III (Greatest Common Divisor) using a function. Complete the codes for the function `gcd` by filling in expressions for `BASE-VALUE`, `BASE-CASE-CONDITION` and `GENERAL-CASE` below:

```
def gcd( a, b )    # assume a >= b
    return BASE-VALUE if BASE-CASE-CONDITION else GENERAL-CASE
```

Test your program with different test cases. Besides submitting the program, attach also screen shots for different test cases to your answers.

Task 5 (for CS453 only)

Re-do Task 2 of Homework 3 using a function `decValue(bit_list)` which returns the equivalent decimal value for the given `bit_list`.

Instead of popping the 0-th element from `bit_list`, we suggest to pop off the last element by the command `bit_list.pop()`.

Consider an example `bit_list` of `[1,0,1,1]`. Applying `bit_list.pop()` returns the last bit 1. At the same time, `bit_list` is shortened to `[1,0,1]`. By calling `decValue(bit_list)`, we obtain 5. Next by multiplying 5 with 2, followed by adding with the last bit 1 that we extracted before, we obtain the decimal value of 11 for the given bit list of `[1,0,1,1]`.

Your job is to fill out the following template:

```
def decValue( bit_list )
    return BASE-VALUE if BASE-CASE-CONDITION else GENERAL-CASE
```

The base case condition is when the bit list is empty.

Test your program with different test cases. Besides submitting the program, attach also screen shots for different test cases to your answers.