COP 2930 - Individual Programming Assignment #5

Due date: Please consult WebCourses for your due date/time

Objectives

1. To give students practice designing programs with lists and sets.

Problem : Card Collections (cards.py)

In class we talked about collecting packs of cards which form a set. For the purposes of this problem, the full set of cards is numbered 0 to n-1. In addition, each card has a value, in cents. Cards may have different values. Each pack of cards has k cards, some of which may be duplicates. Since we only care about new cards, we immediately give all duplicates away.

Write a program that reads in from the user the number of baseball cards in the set (n), the values of each of those n cards, and the contents of three packs of cards.

Your program should make the following computations:

(1) Calculate the total value of all the unique cards in the three packs all together.

(2) Determine which two packs of the three, taken together, create the most value, and print out that value.

Implementation Restrictions

In order to earn full credit, you must store each pack as its own set. Furthermore, you must write a function that takes in both a set of cards and a full list of the values of all cards, and returns the total value of the set. Here is the function protocol:

```
# Pre-conditions: setOfCards is a set of integers in between
# 0 and n-1, where n is the length of the list
# listOfCardValues, and listOfCardValues[i] is the value of
# card number i.
# Post-condition: Returns the total value of the cards
# specified by the set setOfCards.
def totalValue(setOfCards, listOfCardValues):
```

Input Specification

Note: It is guaranteed that whoever uses your program will adhere to these specifications. This means that you do NOT have to check for them!

The number of cards for the whole set, *n*, will be in between 5 and 20, inclusive.
All card values will be in between 1 and 100 cents, inclusive.
The number of cards in each pack, *k*, will be at least 2 and less than *n*.
All card numbers will be in between 0 and *n*-1, inclusive.

Output Specification

The first line of output should be of the form:

```
The total value of all three packs is C cents.
```

where C is the value of all three packs, in cents.

The second line of output should be of the form:

```
The two packs with most value are X and Y worth C cents.
```

where X and Y are (1 and 2, 1 and 3, or 2 and 3) and C is the total value of those two corresponding packs. It is guaranteed that the correct answer will be unique.

Output Sample

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**.

Sample Run

How many cards in the set?
8
List each value, in order of card number, in cents, 1 per line.
5
3
12
6
2
9
1
20
How many cards in a pack?
2
Which cards are in pack 1?
0
5
Which cards are in pack 2?
0
3
Which cards are in pack 3?
6
7
The total value of all three packs is 41 cents.
The two packs with most value are 1 and 3 worth 35 cents.

Restrictions

Please IDLE 3.6 (or higher) to develop your program. Turn in a single file, **cards.py**, representing your solution to the problem.

Your program should include a header comment with the following information: your name, course number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

In addition, note that you can only receive full credit if you follow the implementation restriction previously stated and write your code efficiently.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness
2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.
3) Following the given implementation restrictions.
4) Compatibility to IDLE.