Your final score is circled in orange or blue. Below are the solutions for Versions A and B of Exam #3 along with the regrading policy.

1 Regrading Policy

If there are any grading errors related to your exam, you must notify me in writing by Monday, December 12, 2011. After December 12th, no changes to exam grades will be considered. Below, are the steps that you must follow if you want your exam regraded.

1. Write a formal statement that specifies clearly the error in question.

2. Attach your statement to your exam.

3. During class or office hours, give me your statement along with your exam to reconsider.

Finally, if your grading error is related to wanting to receive more partial credit (on question 8 for example), then your exam will be returned back to you. However, if there is an actual error (e.g., a correct solution is marked incorrect, your exam score is not tallied correctly), then please follow the above steps to have your exam regraded.

2 Version A

1.
   a) True
   b) False
   c) False
   d) False
   e) False

2.
   a) 110010001
   b) 75
   c) 00011111
   d) 4A
3.
   a) cat  
   b) catcat  
   c) 30  
   d) hahahaha

4.
   a) [2, -4, 6]  
   b) [[1, 1], [-2, 4], [3, 9]]

5.
   a) 2  
   b) 2  
   c) 4  
   d) 2

6.
   a) 4  
   b) 11  
   c) 13  
   d) True  
   e) False  
   f) 5

7.
   a) Output of g3: 14  
   Output of g2: 9  
   The value of a is: 10  
   b) b  
   c) a, c  
   d) a  
   e) a  
   f) None
8. This is one of many possible solutions.

Listing 1: q8.py

```python
import random

def roll_dice(n):
    ''' Rolls n dice and returns the result as a list. '''
    dice = []
    for i in range(n):
        dice += [random.randint(1,6)]
    return dice

def main():
    # Get input from user.
    num_dice = int(input("Please enter the number of dice you want: "))
    reps = int(input("Please enter the number times you want to roll the dice: "))

    # Populate dictionary.
    my_dict = {}
    for i in range(reps):
        key = sum(roll_dice(num_dice))
        my_dict[key] = my_dict.get(key, 0) + 1

    # Print results.
    print "Here are the results."
    print "TOTAL \t FREQUENCY (%)"
    for key in sorted(my_dict):
        print "%d \t %.1f" %(key, float(my_dict[key])/reps * 100)

main()
```

3 Version B

1.
   a) False  
   b) False 
   c) False 
   d) False 
   e) True 

2.
   a) 100111111 
   b) 109  
   c) 00101011 
   d) 52 

3.
   a) cat 
   b) cat
c) 20  
d) hahaha

4.

a) [-6, 4, 2]  
b) [[-3, 9], [2, 4], [1, 1]]

5.

a) 3  
b) 3  
c) 6  
d) 3

6.

a) 6  
b) 5  
c) 10  
d) False  
e) True  
f) 7

7.

a) Output of h3: 16  
Output of h2: 10  
The value of c is: 15

b) b  
c) a, c  
d) c  
e) c  
f) None

8. See solution for question 8 on Exam A.