

NAME: _____

CS 111 Exam #1 – Winter 13

Directions:

You have 65 minutes to complete this exam. Please show your work, and carefully indicate your answers. Please write your final answers legibly. Don't spend too much time on any one of the questions. You do not need to include comments in any code you write unless you are unsure about the syntax you use, in which case describing the correct algorithm will earn partial credit. You do not need to use `main()` either, however your indentation should be clear. Good luck!



1 (5)	2 (20)	3 (5)	4 (3)	5 (8)	6 (1)	7 (8)	(50pts)

1. (5) Given the following excerpt taken from `help(math)` about the **factorial** function, write a program that properly imports and uses the math module to calculate the factorial of a number. Your program should ask the user for a number and then print the factorial of that number. You do not need to do any error checking, you can assume the user will enter a valid integer.

```
factorial(...)
    factorial(x) -> Integral
    Find x!. Raise a ValueError if x is negative
    or non-integral.
```

Example output (bold designates user input):

```
please enter an integer: 4
Factorial: 24
```

```
import math
num = raw_input("enter a number: ")
num = int(num)
print "The factorial is", math.factorial(num)
```

Could also use

```
from math import *
```

or

```
from math import factorial
```

and then change print line to

```
print "The factorial is", factorial(num)
```

2. (20) Given the following snippets of code, for each write the **value** and **type** of ANS, or write "ERROR" if that snippet would cause an error:

Example: `ANS = 4+3` You would write: **7, int**

a. `ANS = 8/3`
`2.666....65, float`

b. `ANS = 8//3`
`2, int`

c. `ANS = 8%3`
`2, int`

d. `ANS = round(1.576)`
`2, int`

e. `ANS = int(1.576)`
`1, int`

f. `ANS = ord('C')-ord('A')`
`2, int` (note you do NOT have to know what the actual ASCII values are, only that the C and A are 2 letters apart in the alphabet)

g. `s = "abcd"`
`ANS = len(s)`
`4, int`

h. `s = "abcd"`
`ANS = s[1] + s[3]`
`"bd", string`

i. `s = "abcdef"`
`ANS = s[:4]`
`"abcd", string`

j. `s = "abcdef"`
`ANS = s[-2]`
`"e"`

3. (5) Given the following code:

```
for i in range(-10,2,4):  
    i += 2  
    print i
```

- a. What is printed by this code?

-8
-4
0

- b. Write a while loop to do the same thing as the above code, producing the same results. Your loop should be general such that it would still produce the same output as the for loop if I changed any of the given values (i.e. used -4 as the starting range value instead of -10).

Acceptable answers:

```
i = -10  
while (i<2):  
    print i+2  
    i += 4
```

```
i=-10  
while (i<2):  
    i += 2  
    print i  
    i += 2
```

```
i=-8  
while (i<4):  
    print i  
    i += 4
```

Not acceptable answers:

```
while True:  
    print -8  
    print -4  
    print 0  
    break
```

4. (8) Given a string **s**, write a program that prints the sum of all of the **digits** in **s**, ignoring all other characters. More efficient answers will receive more points.

For example given the string "ab45c* d2" the sum would be 4+5+2=11.

Full credit answers:

```
total = 0
for c in s:
    if c.isdigit():
        total += int(c)
print total
```

```
total = 0
for c in s:
    if "0" <= c <= "9":
        total += int(c)
print total
```

```
total = 0
for c in s:
    if ord("0") <= ord(c) <= ord("9"):
        total += int(c)
print total
```

7 of the 8 points (correct but less efficient):

```
total = 0
for i in range(10):
    total += i * s.count(i)
```

5. (8) Write a Python program that inputs an integer N from the user, and outputs the smallest number after 1 that is a divisor of N (meaning the number divides evenly into N, with no remainder), or a message that N is prime if N is in fact prime (meaning there is no divisor greater than 1 except for N itself). More efficient solutions will receive more credit.

Example output (bold designates user input):

```
please enter an integer: 15  
smallest divisor is 3
```

```
please enter an integer: 17  
that number is prime!
```

Best answer:

```
N = raw_input("enter an integer: ")  
prime = True  
i = 2  
while i < (N**.5 + 1) and prime:  
    if N%i == 0:  
        print "smallest divisor is", i  
        prime = False  
        i += 1  
if prime:  
    print "N is prime"
```

Acceptable answer:

```
N = raw_input("enter an integer: ")  
prime = True  
for i in range(2, N**.5 + 1):  
    if N%i==0:  
        print "smallest divisor is", i  
        prime = False  
        break  
if prime:  
    print "N is prime"
```

If import math, could use

```
math.sqrt(N)+1 instead of N**.5+1
```

Lost efficiency points if did not quit after finding lowest divisor, if checked divisors all the way from 2 to N-1 instead of stopping at $\sqrt{N}+1$.

