

COSC 1336 Exam 1 Review Problem **Solutions**

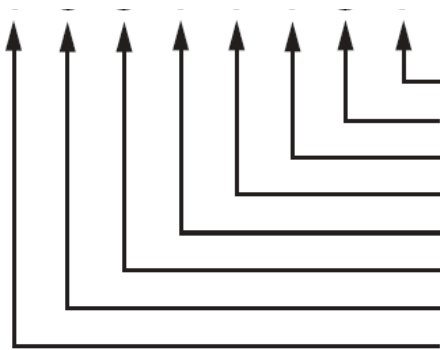
Variables, constants, input/output and processing

Data representations

1. Fill in the chart with the appropriate value if j is the bit position and the value at each position is 2^{j-1}

Bit number:

8 7 6 5 4 3 2 1



1
2
4
8
16
32
64
128

2. Convert the following binary values to decimal

- i. 00000100 4
- ii. 01001010 74
- iii. 01100110 102

3. Convert the following decimal values to one-byte binary numbers. Specify a value for all eight bits in the byte.

- i. 5 00000101
- ii. 105 01101001
- iii. 127 01111111

4. Convert the following binary bytes to their corresponding ASCII characters i.

01110010 r

ii. 00101101 -

iii. 00110000 0

5. Convert the following ASCII characters to their corresponding binary representations. i.

'C' 01000011

ii. '4' 00110100

iii. '@' 01000000

6. Determine whether each of the following Python identifiers is valid or invalid. If it is valid, say whether it is appropriate for a variable or a constant. If it is invalid say why it is invalid, and whether it breaks an interpreter rule or a convention.

1. item#1

invalid. rule. # not allowed

2. data

valid. variable

3. 1variable

invalid. rule. must start with letter or underscore

4. sq_height

valid. variable

5. PAY_RATE

valid. constant

6. num5

valid. variable

7. Circle_Radius

invalid. convention. should start w/
lowercase

8. sq ft

invalid. rule. contains a space

9. bin-2

invalid. rule. contains an operator
(-)

10. DISTANCETOMOON

invalid. convention. separate
words with

7. Compute the value of each of the following expressions. Give the type of the output.

a. $10 + 3$

13 int

b. $-9.4 - 6$

-15.4 float

c. $10.0/3.0$

-
3.3 float

d. $10/3.0$

-
3.3 float

e. $10.0//3.0$

3.0 float

f. $10//3$

3 int

g. $10/3$

-
3.3 float

h. $10 \% 3$

1 int

i. $2 ** 3.0$

8.0 float

j. $4 / 8$

0.5 float

k. $8 / 1.5$

-
5.3 float

l.

8

/

/

1

.

5

5.0 float

8. For each of the following statements, assume that the statement directly follows these lines of code:

```
a = 5 b =  
2
```

What output does each statement produce?

1. `print("a = ", a, ", b = ", b, sep="")`
2. `print("sum:", a+b)`
3. `print("sum:", "a+b")`
4. `print(a // b, "feet")` feet

Simple functions

Problem 9. Write the Python code corresponding to the following pseudocode algorithm for calculating the cost of carpet for a room.

```
read in room length
read in room width
read in price per square foot

calculate room area by multiplying room width by room length
calculate total cost by multiplying room area by price per square foot

display total cost
```

See carpet_price done.py

Selection

10. Given the following variables, for each problem, determine what will be printed on the screen when the code is executed. If there is an error, say what the error is.

```
num1 = 4
num2 =
13.0 tf =
False
```

a.

```
if num2 > 5:
    print("Big")
else:
    print("Small")
```

Big

b.

```
if num1 == 5:
    print("Message 1")
else:
    print("Message 2")
```

Message 2

c.

```
if num1 = 5:
    print("Message 1")
else:
    print("Message 2")
```

This will result in an error because the assignment operator is used rather than the comparison operator in the condition.

d.

```
if num2 > num1:
    print("Tom")
if not tf:
    print("Harry")
```

Tom

e.

```
if tf == num2 > num1:
    print("this tf:", tf)
```

else:
1
s
e
:

```
print("that tf:", tf)
```

in the first line, `tf` which is equal to `False` is first compared to `num2` resulting in a `False` result. `num2 > num1` is executed resulting in `True`, so the whole line has a `False` value resulting in the output:

that
tf:
False

f.

```
tf = True
if tf == num2
> num1:
    print("this tf:", tf)
else:
    print("that tf:", tf)
```

This time, the variable `tf` is first assigned to have the value `True`. Because there are no parentheses around `num1 > num2`, `tf == num2` is evaluated first, resulting in a `False` result or `0`, which is not greater than `num1`, so the whole expression is `false`.

that
tf:
True

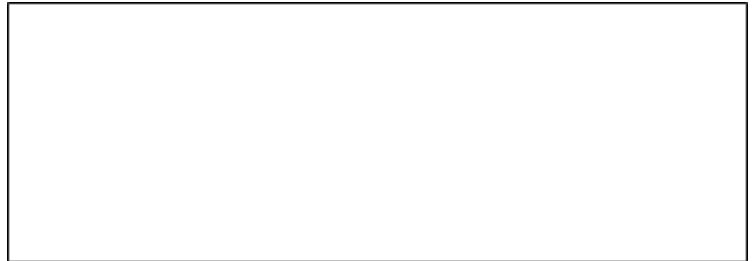
Harry

```
g. tf = "True"
   if tf
     ==
     num2
   >
   num1:
     print("this tf:",tf)
   else:
     print("that tf:",tf)
```

In this case, tf has been changed to be the string value

"True", which is not the same as the Boolean value True. So the equality comparison operator will return False, so we execute the else block. But when we print tf, it will print out True, just like if it were a Boolean value.

t
h
a
t
t
f
:
T
r
u
e



Problem 11. Given the following variables, determine whether each expression is True or False.

```
a =  
1  
b = 5  
c = -1  
my_bool = False;
```

a. `a == -c and a < b`

True

b. `not (not my_bool)`

False

c. `b < c and 2 < b`

False

d. `(not my_bool and (c < b or a == b)) or not my_bool` True

e. `my_bool and ((a < b and c < b) and c < 0)` False

12. Write a loop of the specified type that sums up the even integers between 1 and 21.

a. a while loop

one alternative:

```
sum= 0  
num = 1  
while num < 21:  
    if num%2 == 0:  
        sum += num  
    num += 1
```

another possibility

```
sum = 0  
num = 2  
while num < 21:  
    sum = sum + num  
    num = num + 2
```

b. a for loop

```
sum = 0  
for i in range(2,21,2):  
    sum = sum + i
```

13. What will the following lines of code print out?

a. `num = 5`
`while num < 5:`
 `print(num, end=" ")`
 `num = num - 1`

Nothing - num is never greater than 5

b. `num = 5`
`while True:`
 `print(num, end=" ")`
 `num = num - 1`
 `if num < 5:`
 `num +=1`

This is an infinite loop. This will print out 5 over and over into infinity.

14. Write a set of nested loops that display 10 rows of # characters. There should be 15# characters in each row.

```
for row in range(10):
    for column in range(15):
        print('#', end='')
    print()
```

15. Write code that prompts the user to enter a number in the range of 1 through 100 and validates the input.

```
number = float(input('Enter a positive nonzero number: '))
while number <= 0 or number > 100:
    print('That is an invalid value.')
    number = float(input('Enter a positive nonzero number: '))
print ('Thanks!')
```