A. Short Answer:

(1) What are the values of the following expressions?

\[ \frac{10}{5} + 3 \times 2 \] __________ 8.0

\[ 3 ** (3+ (9\%4) \% 2) \] __________ 27

\[ 1.0/2.5>5-3 \text{ or } 3==1+2 \text{ and } 5*2<2*5 \] __________ False

(2) Explain what the \texttt{str()} function does. Give an example.

\texttt{str()} casts (converts) the input value to a string
e.g. \texttt{str(3) \rightarrow "3"}

(3) Explain the difference between “while” and “for” loops. When should you use one versus the other?

A while loop is a condition controlled loop.
A for loop is a counter controlled loop.
You would use a for loop when you know how many times you want to repeat a block of code, and a while loop when the number of repetitions is unknown (e.g. depends on randomness or the user).
B. Comprehension:

(1) What does the code to the right print?

```python
a = 10
b = 3 * (a-5)
c = b + (b//a) * a
a += c
print("a: "+str(a))
print("b: "+str(b))
print("c: "+str(c))
```

a: 35
b: 15
c: 25

(2) What does the code to the right print?

```python
a=0
while a<10:
    print(a)
    if a%2==0:
        a=a+1
    elif a==6:
        a=a-1
    else:
        a=a*2
print(a)
```

0 1 2 3 6 7 14

(3) What does the code to the right print?

```python
def fun(x):
    z = x
    for i in range(3,x):
        z += i
    return z

a=2
b=7
if a*2 < 10 and not b==a+5:
    a = fun(a+b)
if not (1<a or 3>b):
    b = fun(a*b)
elif b<=a and b//a+b%a<10*(b-a):
    a = fun(5-a)
else:
    b = fun(8-a)
print(a)
print(b)
```

2 18
C. Programming:

(1) Write a program that asks the user to enter two positive integers. Your program should loop from the lower number to the higher number and sum all of the values not divisible by 5. Print the final sum. Eg. If the user enters the numbers 13 and 4 your program would sum 4+6+7+8+9+11+12+13.

```python
a = int(input("Enter a number: "))
b = int(input("Enter a number: "))
if a>b:
    temp = b
    b = a
    a = temp
sum=0
for i in range(a,b+1):
    if i % 5 != 0:
        sum+=i
print(sum)
```

(2) Write a program that repeatedly asks the user to enter words until they enter a period ("."). Your program should then print all of the words they entered (including the period) on a single line separated by spaces. E.g.: Enter some words (. to stop):
> hello
> world
> .
hello world.

```python
output=''
s = ''
print("Enter some words(. to stop): ")
while s!=".":
    s = input("> ")
    output += s+" 
print(output)
```
(3) Complete the code below to create a function that takes an integer argument n and displays a pattern like the examples below. If the input number is not at least 2, your program should print a small error message instead of drawing.

E.g. square(5)  E.g. square(3)  E.g. square(1)
#####  ###  Input is too small.
###  #  ###
#  #  #  #
#  #  #  #  #
###  #  #####

```python
def square(n):
    if n<=1:
        print("Input is too small")
        return
    # top
    row=''
    for i in range(0,n):
        row+='#'
    print(row)
    # sides
    for i in range(0,n-2):
        row='#
        for j in range(0,n-2):
            row+=' '
        row+='#
        print(row)
    # bottom
    row=''
    for i in range(0,n):
        row+='#'
    print(row)

# alternate solution string multiplication...
if n<2:
    print("input is too small.")
else:
    print("#"*n)
    for i in range(n-2):
        print("#"+ "*"+(n-2)+"#")
    print("#"*n)
```
(Extra practice) Complete the code below to create a function that takes in an integer argument n, and displays a series of numbers as follows: for every positive integer less than n, display the sum of the values from that number to n, followed by the sum of all of the displayed sums. You may not use the built-in sum function for this problem. (Note, the comments in the examples below are not part of the output).

**E.g.** sumsFunc(6)  
21  # =1+2+3+4+5+6  
20  # = 2+3+4+5+6  
18  # = 3+4+5+6  
15  # = 4+5+6  
11  # = 5+6  
6   # = 6  
91  # =21+20+18+15+11+6

```python
def sumsFunc(n):
    total=0
    for i in range(1,n+1):
        rowSum = 0
        for j in range(i,n+1):
            rowSum+=j
        print(rowSum)
        total+=rowSum
    print(total)
```

**E.g.** sumsFunc(3)  
21  # =1+2+3  
20  # = 2+3  
18  # = 3  
15  # = 3  
14  # =6+5+3  
11  # = 3  
6   # = 6  
91  # =21+20+18+15+11+6