

CENG 1003 Midterm Examples

Q1

Give the output (results of the print statements) of the following code segment:

```
a = 2
b = 3
c = a + b;
b = b * c;
if (a+b > 10):
    b *= 2
else:
    c += 2
print(a, b, c)
b = 8
c = a + b + c
a = b % (a+2)
b = c
c = b
b -= 1
a *= b
print(c-1, a+2*b, a)
```

Q2

Give 8 different calls of this function (i.e. 8 different pair of values for a and b) where each call will result in printing a different message.

```
def q2(a,b):
    if (a > b):
        if (a < 10 and b < 12):
            if ((a+b) % 2 == 1): print("Answer1")
            elif (a+b > 10): print("Answer2")
        else:
            print("Answer3")
    else:
        if (a >= 10 or b >= 7): print("Answer4")
        elif (a > 4): print("Answer5")
        elif (a != b):
            if (a <= 2): print("Answer6")
            else: print("Answer7")
        else: print("Answer8")
```

Use the following table for your answer:

CALL	OUTPUT	CALL	OUTPUT
q2(____, ____)	Answer1	q2(____, ____)	Answer5
q2(____, ____)	Answer2	q2(____, ____)	Answer6
q2(____, ____)	Answer3	q2(____, ____)	Answer7
q2(____, ____)	Answer4	q2(____, ____)	Answer8

Q3

Give the output of the following code segment:

```
a = 0
b = 0
while (a+b < 20):
    a = a + 5
    print(a, b)
    b = b + 2
```

Q4

Give the output of the following code segment:

```
for n in range(5,10):
    print("XXX", end=":")
    for c in ["A", "B", "C"]:
        print(c,end=".")
        if n<8: print("?",end="")
        else: print("!", end="")
    print(n)
print("YYY")
```

Q5

Consider the following function definition :

```
def f(a,b):
    while (a != b):
        if (a<b): b = b - a
        else: a = a - b
    return a
```

First try to understand how the function computes its return value than give what the following code segment will output:

```
a = 12
b = 9
print("Answer 1:", f(a,b))
a = f(a,b)
print("Answer 2:", f(17+a,b*4), f(50, f(90, 60)))
```

Q6

Give a code segment where the user is asked to enter an animal name which has at least 5 letters. You need to keep asking until the user gives a string with at least 5 characters. Then ask for another animal name which has more letters than the first one. Again you need to keep asking until the second string has more characters than the first one. (**Hint:** You can use the `len` function to return the length of a string: `len("abcd")` will return 4.)

Q7

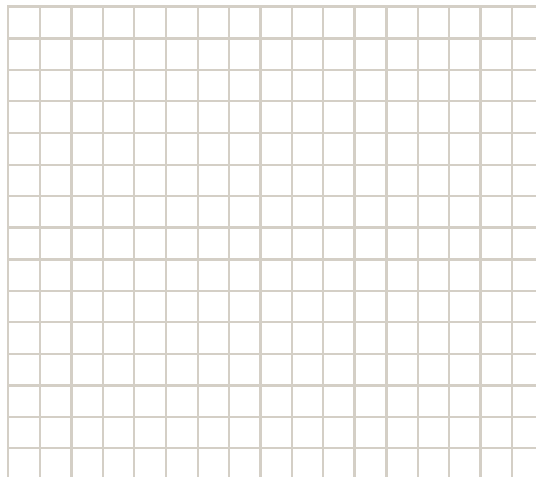
Give the output of the following code segment:

```
list1 = [ 2, 7, 5, 8, 9 ];
list2 = [ 1, 2, 3, 11, 12, 13, 15, 21, 22, 23, 32]
t = 0
k = 0
for i in list1:
    for j in list2:
        if j%i==0:
            t = t + (j/i)
        elif i > j:
            k = k + 1
    print(t)
print("k =",k)
```

Q8

Assuming alex is a turtle variable with heading 0 (to the East) give an approximate drawing of the following code's result:

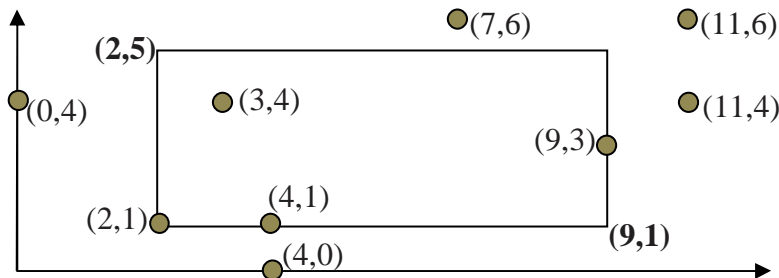
```
dist = 100
gap = 15
angle = 90
for i in range(1,10):
    alex.forward(dist)
    alex.right(angle)
    if (i%3 == 0):
        alex.penup()
        alex.forward(gap)
        alex.right(angle)
        alex.forward(gap)
        alex.pendown()
        dist -= 2*gap
        angle *= -1
```



Hint: alex.right(-30) is equivalent of alex.left(30). You may give your answer on the above grid by assuming each cell is 10x10 pixels. You may put alex anywhere you want initially.

Q9

Write a function called `query_point` that will take 6 arguments $(x1, y1, x2, y2, x3, y3)$ where $(x1, y1)$ is the top left coordinate of a rectangle, $(x2, y2)$ is its bottom right corner and $(x3, y3)$ is the coordinates of a query point. Your function will determine if the query point $(x3, y3)$ is inside the defined rectangle, outside of it, or on one of its edges and print a relevant message. See the picture and the example calls for further explanation:



Example calls:

```
query_point(2,5, 9,1, 3,4)  should return "inside"
query_point(2,5, 9,1, 4,1)  should return "on edge"
query_point(2,5, 9,1, 7,6)  should return "outside"
query_point(2,5, 9,1, 9,3)  should return "on edge"
query_point(2,5, 9,1, 11,4) should return "outside"
query_point(2,5, 9,1, 0,4)  should return "outside"
query_point(2,5, 9,1, 4,0)  should return "outside"
query_point(2,5, 9,1, 11,6) should return "outside"
query_point(2,5, 9,1, 2,1)  should return "on edge"
```

Q10

Write a function called `middle` that will take 3 numbers and return the middle number among these three. If any two of the parameters are same your function should return the string "error".

Example calls:

```
middle(4,7,4)  should return "error"
middle(3,5,5)  should return "error"
middle(2,2,8)  should return "error"
middle(9,9,9)  should return "error"
middle(3,5,8)  should return 5
middle(9,7,4)  should return 7
middle(4,1,6)  should return 4
middle(5,8,2)  should return 5
middle(9,3,8)  should return 8
middle(2,4,3)  should return 3
```

Q11

Describe what the following function computes. Don't give a line by line explanation of what the function does but explain how the return value depends on the input value (i.e. this function finds the largest number in a list of numbers).

```
def f(numbers):  
    s = 0  
    for n in numbers:  
        s = s + n  
    a = s / len(numbers)  
    p = 0  
    for n in numbers:  
        if n>a:  
            p = p + 1  
    return p
```

Q12

By filling in the spaces for the given function definition, make it to return the factorial of its given parameter n.

```
def fact(n):  
    r = ____  
    while (n>=____):  
        r = ____ * ____  
        n = n - ____  
    return ____
```

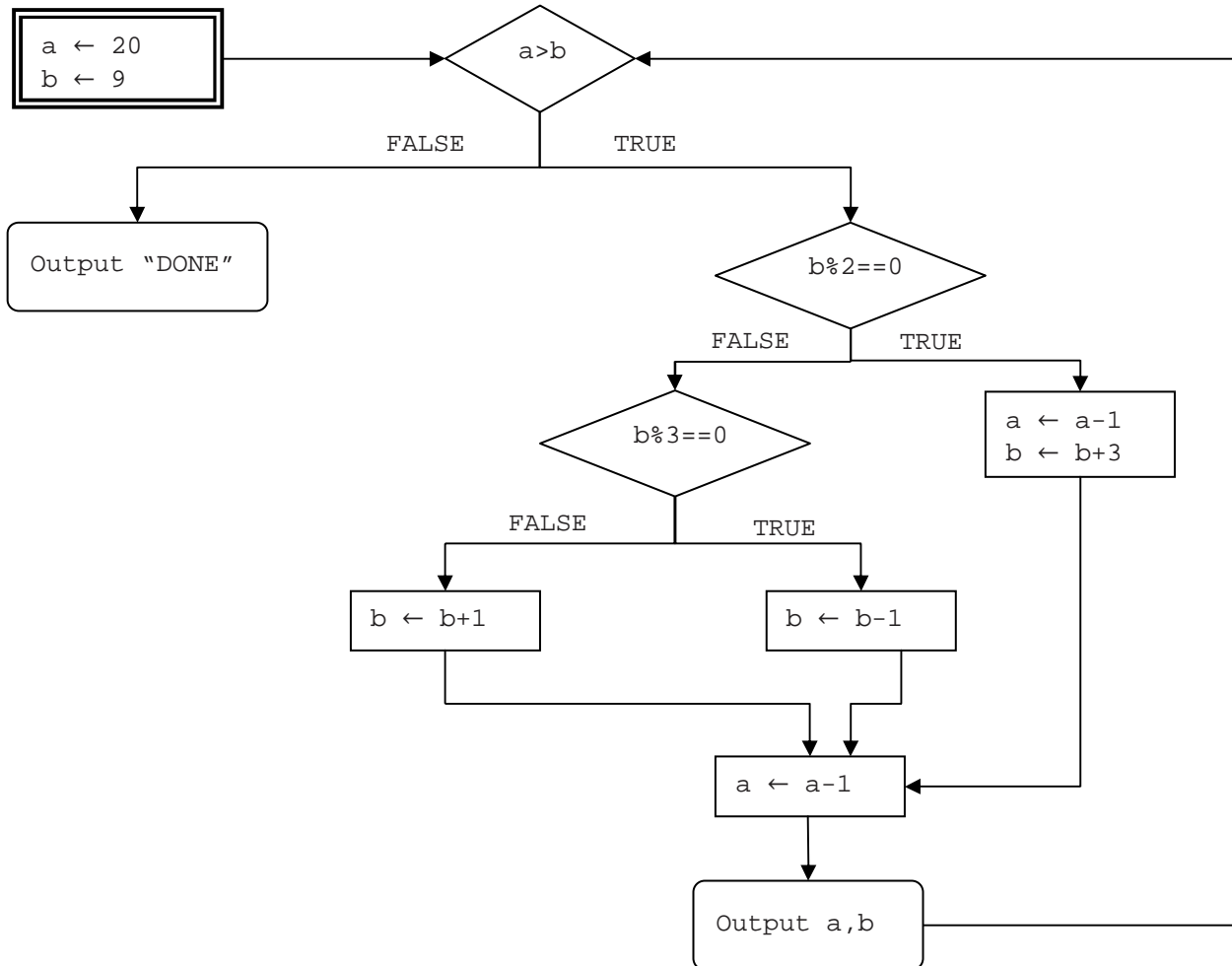
Q13

For the expressions at left give their value, for the expressions at right add parantheses such that expression's value becomes the value given.

Expression	Value		Expression	Value
$5*3-2**2$			$20\%5+2*3$	18
$24/6*2$			$48/10-2-57//8+2$	1
$10*(1+(1+3)**(12\%5)-2)$			$7-4*3-8-9$	10

Q14

Convert the following flowchart into actual code. The arrow notation (\leftarrow) means assignment. The diamonds refer to conditional tests.



If you successfully converted the flowchart into equivalent Python code when you run it, the output should be:

```
19 8
17 11
16 12
14 15
DONE
```