

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

## CSCI-UA.0002 – While Loops and Boolean Expressions

1. Write the output of each program in the space adjacent to it. If there is an error, indicate that an error will occur.

```
1. x, y = 0, 0
   while x <= 2 or y <= 4:
       x += 2
       y += 2
       print(x, y)
2 2
4 4
6 6
```

```
2. # assume that the user types in 2 when prompted for input
   start = input('plz enter a number\n>')
   number = start
   while number < 10:
       print(number)
       number += 3
```

**Error (TypeError, comparing str to int in while loop condition)**

```
3. number = 1
   while number < 5:
       if number > 3:
           print('%s is too big' % (number))
       else:
           print(number * 'hi')
       number += 1
hi
hihi
hihihi
4 is too big
```

```
4. x = 0
   while x != 5:
       print(x)
       x += 2
infinite loop
```

2. Fill in the blanks in the program below.

It will continue to ask for numbers as long as the number entered meets one of the following conditions:

1. it's even
2. it's equal to 7

Once the user enters a number that does not meet the above conditions, stop asking for numbers and then print out all of the numbers that have been entered. See example ^

```
# Example Output
give me an even number (7 is ok too)
>62
give me an even number (7 is ok too)
>7
give me an even number (7 is ok too)
>8
give me an even number (7 is ok too)
>123
62 7 8
```

```
all_numbers = ''
answer = int(input('give me an even number (7 is ok too)\n>'))
while ___answer == 7 or answer % 2 == 0_____:
    all_numbers ___all_numbers + str(answer) + ' '____
    answer = int(input('give me an even number (7 is ok too)\n>'))
print(all_numbers)
```

3. Use DeMorgan's Laws and logical opposites to simply the boolean expression below:

```
while not (num < 1 or num > 6): # simplify this while loop's condition
while num >= 1 and num <= 6
```