CS 112 – Summer 2022
Midterm Review - Exercises

Part 1: Questions 1 to 5 T/F questions

1. `(x * y) / 2` is an example of an expression

   A. `True`
   B. `False`

2. The following code prints `[2,3,4]` when is run

```
xs = [1, 2, 3, 4, 5]
print(xs[-4:4])
```

   A. `True`
   B. `False`

3. After the following code is executed, the final value of `x` is `32`

```
x = 15
x = x + 1
x = x * 2
x = 30 - x
```

   A. `True`
   B. `False`

4. The following code prints `23rd` when the following code is executed

```
day = 23
if day % 10 == 1:
    ending = "st"
elif day % 10 == 2:
    ending = "nd"
elif day % 10 == 3:
    ending = "rd"
else:
    ending = "th"
print(str(day) + ending)
```

   A. `True`
   B. `False`

5. The body of the loop below will execute `7` times

```
number = 70
guess = 55
while number != guess:
    if number > guess:
        guess = guess + 10
    else:
        guess = guess - 1
print('The number is:', guess)
```

   A. `True`
   B. `False`

Part 2: Questions 6 to 20 Multiple Choice questions

6. After the program runs, what is the value of `y`?

```
def print_sum(num1, num2)
    print(num1 + num2)
y = print_sum(4, 5)
```

A. `4 5`
B. `9`
C. `45`
D. `None`

7. Which of the following identifiers is valid?

A. `max_age`
B. `32area`
C. `transfer$`
D. `True`

8. What is the value of `x` after the following code is executed?

```
x = 17
if x * 2 <= 34:
    x = 0
else:
    x = x + 1
x = x + 1
```

A. `1`
B. `18`
C. `19`
D. `35`

9. A child is required to use a booster seat in a car until the child is `9` years old, unless the child reaches the height of `59` inches before age `9`.
Which expression can be used to decide if a child requires a car seat or not?

A. `if age < 9 or height < 59:`
B. `if age >= 9 or height >= 59:`
C. `if age >= 9 and height >= 59:`
D. `if age <= 9 and height <=59:`

10. Grover Cleveland served as president of the United States from `1885` to `1889` and from `1893` to `1897`. Which expression correctly detects this range?

A. `(1885 < x < 1889) or (1893 < x < 1897)`
B. `(1885 <= x <= 1889) or (1893 <= x <= 1897)`
C. `(1885 <= x <= 1889) and (1893 <= x <= 1897)`
D. `(1885 < x <= 1889) or (1885 < x <= 1889)`

11. Excess indentation must be removed from which lines to make the code correct?

```
1. print('start')
2.     if x > 10:
3.         print('large')
4.         else:
5.     print('small')
6. print('done')
```

A. 1, 6
B. 1, 2, 3
C. 2, 3, 4
D. 2, 4, 5

12. What is the output?

```
count = 0
while count < 3:
     print('loop', end=' ')
count = count + 1
print('final value of count:', count)
```

A. loop final value of count: 1
B. loop loop loop final value of count: 3
C. loop loop loop loop final value of count: 4
D. loop loop loop loop … #this answer means an infinite loop

13. How many times does the while loop execute for the given input values of `-1 4 0 9`?

```
user_num = 3
while user_num > 0:
    # Do something
    user_num = int(input())
```

A. 0
B. 1
C. 2
D. 3

14. What sequence is generated by `range(1, 10, 3)`?

A. 1 4 7
B. 1 11 21
C. 1 3 6 9
D. 1 4 7 10

15. Which `range()` function call generates every even number between 20 and 30 (including both 20 and 30)?

A. range(20, 30, 2)
B. range(20, 31, 2)
C. range(30, 20, 2)
D. range(20, 22, 24)

16. Which expression using parentheses is equivalent to the following expression:

```
x -y * -z / 3
```

A. `(x -y) * ((-z) / 3))`
B. `x -((y * (-z)) / 3)`
C. `x -(y * ((-z) / 3))`
D. `(x -(y * (-z))) / 3`

17. Which print statement would display: `I won't quit!`

A. `print('I won\\'t quit!')`
B. `print('I won't quit!')`
C. `print('I won\'\t quit!')`
D. `print('I won\'t quit!')`

18. What is the output?

```
num = 10;
while num <= 15:
    print(num, end=' ')
    if num == 12:
        break
    num += 1
```

A. `10`
B. `10 11`
C. `10 11 12`
D. `10 11 12 13 14 15`

19. What is the output?

```
for i in range(11):
    if i == 6:
        continue
    else:
       print(i, end=' ')
```

A. `0 1 2 3 4 5`
B. `0 1 2 3 4 5 6`
C. `0 1 2 3 4 5 7 8 9`
D. `0 1 2 3 4 5 7 8 9 10`

20. What is output?

```
new_list = ['python', 'development']
new_list.append('in progress')
print(new_list)
```

A. `['python', 'development', 'in progress']`
B. `['python', 'development', ['in progress']]`
C. `['python', 'in progress']`
D. `['python', 'developmentin progress']`