Question 1:
Guess the output of this program:

```
1. print('abc dbe'.split('b'))
```

- ['a', 'cd', 'e']
- ['a', 'c d', 'e']
- ['a', 'c', 'd', 'e']

Question 2:
Guess the output of this program:

```
1. print('abc dbe'.split('db'))
```

- ['abc ', 'e']
- ['abc e']

Question 3:
Guess the output of this program:

```
1. print('abc dbe'.split('dbx'))
```

- 'abc dbe'
- ['abc dbe']

Question 4:
Guess the output of this program:

```
1. print(','.join('A#B#C'.split('#')))
```

- A#B#C
- A,B,C
- Error

Question 5:
For input:

5,,,2, 3

```
1. items = tuple(sorted(map(int, input().replace(',', ' ').split())))
2. print(items[0]*items[1] + items[2])
```

What is the output?

- 17
- 13
- 11

Question 6:
Guess the output of this program:

```
1. print('{}*{} = {}', 2, 3, 2*3)
```

- 2*3 = 6
- Something else

{}*{} = {} 2 3 6 .format is missing

Question 7:
These 2 lines will print the same results?

```
1. print('%d*%d = %d' % (2, 3, 2*3))
2. print('{}*{} = {}'.format(2, 3, 2*3))
```

- True
- False

Question 8:
Guess the output of this program:

```
1. num1, num2 = 2, 3
2. res = num1 * num2
3. print('{num1}*{num2} = {result}'.format(num1=num1, num2=num2, res = result))
```

- Error
- 2*3 = 6

```
print('{num1}*{num2} = {result}'.format(num1=num1, num2=num2, result = res))
```

Question 9:
Guess the output of this program:

```
my_lst = ['10', 3]
print('{lst[0]} * {lst[1]}'.format(lst = my_lst))
```

- 30
- 10 * 3
- '101010'

Question 10:
Guess the output of this program:

```
i = 15
string = '{:4}|{:<8}|{:^6}'.format(i, i* i * i * i, i * i * i)
print(string.replace(' ', '*'))
```

- **15|***50625|*3375*
- **15|50625***|*3375*
- **15|*50625**|*3375*

Question 11:
Guess the output of this program:

```
val = 13.123216789
print('{:12.4f}'.format(val).replace(' ', '*'))
```

- *****13.1232
- ***13.1232

- ***13.123217

Question 12:
Guess the output of this program:

```
class Student:
    def __init__(self, name, id):
        self.name, self.id = name, id

    def __repr__(self):
        return f'Student({self.name}, {self.id})'

most = Student('mostafa', '1112KS1')
print(f'{most}')
```

- Student(mostafa, 1112KS1)
- <__main__.Student object at 0x7fc58fb59390>

Question 13:
Guess the output of this program:

```
print('{0}{0}{0}'.format(1, 2, 3))
```

- 123
- 111

Question 14:
Guess the output of this program:

```
string = 'Hey'
string[1] = string[1].upper()
```

- HEy
- Error

string is immutable

Question 15:
Background:

- is newline handled the same in different OSes? No, due to historical reasons. Here is an initial informal specification

- \n is used for linux

- \r\n for windows

- \r for (old) mac

- Reading: https://stackoverflow.com/questions/1761051/difference-between-n-and-r

```
1. print('H\rey\r\nHow\n\n\n\nAre\tyou?'.splitlines())
```

- ['H', 'ey', 'How', '', '', '', 'Are\tyou?']
- ['H', 'ey', 'How', 'Are\tyou?']
- ['Hey', 'How', 'Are\tyou?']

Question 16:
Guess the output of this program:

```
1. print(f"Hey {'\n'} Mostafa")
```

- Error
- Hey \n Mostafa
- Hey
  Mostafa

SyntaxError: f-string expression part cannot include a backslash

Question 17:
Guess the output of this program:

```
1. line = '\n'
2. print(f"Hey {line} Mostafa")
```

- Hey

Mostafa

- Hey \n Mostafa

Question 18:
Background: Python **raw string** is created by prefixing a string literal with ‘r’ or ‘R’. Python raw string treats backslash (\) as a literal character. This is useful when we want to have a string that contains backslash and don’t want it to be treated as an escape character.

Guess the output of this program:

```
print(r"Hey \n Mostafa")
```

- Hey \n Mostafa
- Hey
  Mostafa

Question 19:
The output of this program is:

mostafa has salary 100

?

```
class Employee:
    def __init__(self):
        self.name = 'mostafa'
        self.salary = 100

    def __str__(self):
        return '{0.name} has salary {0.salary}'.format(self)

print(Employee())
```

- True
- False

Question 20:
Guess the output of this program:

```
1. string = 'hello world'
2. print(string.endswith(("wor", "rld", 'hello '), 6))
```

Consider the endswith documentation

```
1. def endswith(self, suffix, start=None, end=None):
2.     """
3.     S.endswith(suffix[, start[, end]]) -> bool
4.
5.     Return True if S ends with the specified suffix, False otherwise.
6.     With optional start, test S beginning at that position.
7.     With optional end, stop comparing S at that position.
8.     suffix can also be a tuple of strings to try.
9.     """
```

- True
- False
- Error

We can pass multiple strings to check at once in a tuple. The used S at position 6 is world

Question 21:
Guess the output of this program:

```
1. print("{:07}".format(123))
```

- Error
- 0000123
- 123

Question 22:
Guess the output of this program:

```
1. print('{:011.1f}'.format(123.17))
```

- 123.1700000
- 0123.170000
- 000000123.2
- 000000123.1