

```
def mystery(a=1, b=3, c=6):  
    print(a + 2 * b + c)
```

What will the above code print when invoked as `mystery(10, 2)`?

- A. 11
- B. 13
- C. 20
- D. 21
- E. 22

Answer: C

`mystery(10, 2)` is equivalent to `mystery(10, 2, 6)` where the third argument `c` is using the default value specified in the function header.

```
def mystery(a=1, b=3, c=6):  
    print(a + 2 * b + c)
```

What will the above code print when invoked as `mystery(b=7)`?

- A. 11
- B. 13
- C. 20
- D. 21
- E. 22

Answer: D

`mystery(b=7)` is equivalent to `mystery(1, 7, 6)` with a result of 21

```
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

end: string appended after the last value, default a newline.

flush: whether to forcibly flush the stream.

The docstring for print is above. Which of the following expressions would print "a,b,c"?

- A. `print("a", "b", "c")`
- B. `print("a", ", ", "b", ", ", "c")`
- C. `print("a", end=",")`
`print("b", end=",")`
`print("c", end=",")`
- D. `print("a", "b", "c", end=",")`
- E. `print("a", "b", "c", sep=",")`

Answer: E

Answers A, B and D would have (extraneous) spaces (because of the default value for sep), C is close but would have a trailing comma.