Path testing : Question 1

↳ Tour all the possible outcomes.

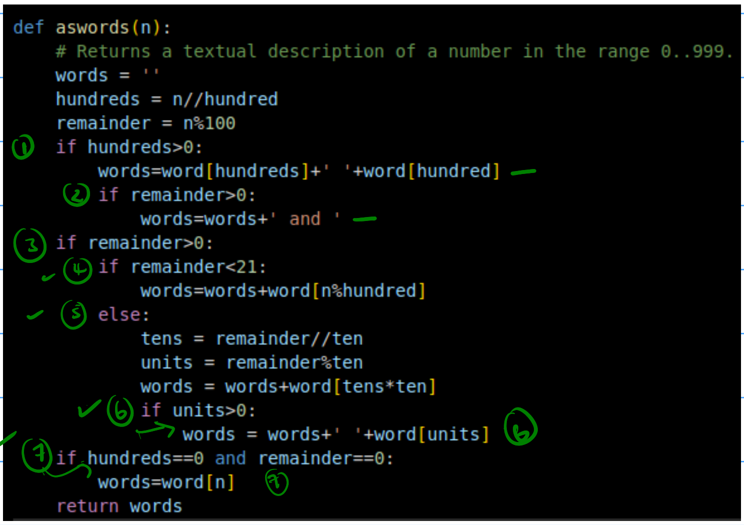
Example:

```
def function(n):
    if n < 0:
        return n      ← ①
    for i in range(n - 2):   ← ②
        something()   ← ③
    return 2      ← ④
```

All paths:

①, ②, ③, ④

- Make 8 path coverage cases



```
def aswords(n):
    # Returns a textual description of a number in the range 0..999.
    words = ''
    hundreds = n//hundred
    remainder = n%100
①   if hundreds>0:
        words=word[hundreds]+' '+word[hundred]
②       if remainder>0:
            words=words+' and '
✓ ③ if remainder>0:
✓ ④     if remainder<21:
            words=words+word[n%hundred]
✓ ⑤     else:
            tens = remainder//ten
            units = remainder%ten
            words = words+word[tens*ten]
✓ ⑥         if units>0:
                words = words+' '+word[units]   ⑥
✓ ⑦ if hundreds==0 and remainder==0:
        words=word[n]   ⑦
    return words
```

Test 1:

input = 0

path = ⑦

Test 2:

input = 10

path = ③ → ④

Test 3:

input = 25

path = ③ → ⑤ → ⑥

Test 4:

input = 20

path = ③ → ⑤

Test 5:

input = 100

path = ①

Test 6:

input: 105

path: ① → ② → ③ → ④

Test 7 : 125

path: ① → ② → ③ → ⑤ → ⑥

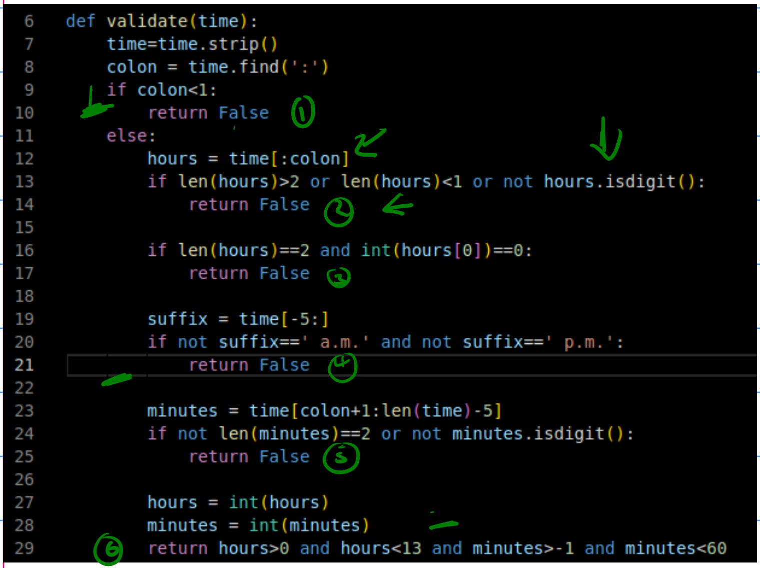
Test 8:

input: 130

path: ① → ② → ③ → ⑤

Statement coverage: Question 2

$\boxed{00}$: 00 _ p.m.



```
 6  def validate(time):
 7      time=time.strip()
 8      colon = time.find(':')
 9      if colon<1:
10          return False
11      else:
12          hours = time[:colon]
13          if len(hours)>2 or len(hours)<1 or not hours.isdigit():
14              return False
15
16          if len(hours)==2 and int(hours[0])==0:
17              return False
18
19          suffix = time[-5:]
20          if not suffix==' a.m.' and not suffix==' p.m.':
21              return False
22
23          minutes = time[colon+1:len(time)-5]
24          if not len(minutes)==2 or not minutes.isdigit():
25              return False
26
27          hours = int(hours)
28          minutes = int(minutes)
29          return hours>0 and hours<13 and minutes>-1 and minutes<60
```

Test 1

input = "12"

statement: (6) → (7) → (8) → (9) → (10)

Output: False

Test 2:

input: "hh:00"

output: False

Test 3

input: "111 :00"

output: False

Test 4:

input: "11:00_pm"

output: False

Test 5:

input: "11:m _p.m."

Output: False

Test 6:

input: "09:23_p.m."

Output: True