

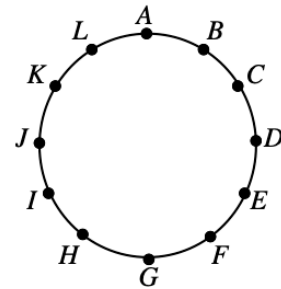
Cayley-1997-06-popping-ballons

July 25, 2020

1 Cayley Contest, 1997, Question 6

6. Twelve balloons are arranged in a circle as shown. Counting clockwise, every third balloon is popped, with *C* the first one popped. This process continues around the circle until two unpopped balloons remain. The last two remaining balloons are

(A) *B, H* (B) *B, G* (C) *A, E*
(D) *E, J* (E) *F, K*



We can represent the set of remaining balloons as a string by listing their letters in clockwise order, starting with *A*. Initially, all balloons remain.

```
[1]: init_balloons = 'ABCDEFGHIJKL'
```

We are told to pop every third balloon, until only two remain. We skip over the first two balloons using Python string slicing.

```
[2]: balloons = init_balloons
front = balloons[:2]
print('front = ', front)
```

```
front = AB
```

We pop the third balloon.

```
[3]: pop = balloons[2]
print('pop = ', pop)
```

```
pop = C
```

We grab the rest of the balloons using string slicing.

```
[4]: rest = balloons[3:]
print('rest = ', rest)
```

```
rest = DEFGHIJKL
```

Since the balloons are arranged in a circle, the front now appears after the rest.

```
[5]: balloons = rest + front
     print('balloons =', balloons)
```

```
balloons = DEFGHIJLAB
```

Now repeat the above steps until only two balloons remain. Put the above procedure in a function.

```
[6]: def pop_balloons(balloons):
     while len(balloons) > 2:
         front = balloons[:2]
         pop = balloons[2]
         print('from ', balloons, ' pop', pop)
         rest = balloons[3:]
         balloons = rest + front

     return balloons
```

Finally run this function on the initial set of balloons.

```
[7]: remain = pop_balloons(init_balloons)
     print('remain =', remain)
```

```
from ABCDEFGHIJKL pop C
from DEFGHIJLAB pop F
from GHIJLABDE pop I
from JKLABDEGH pop L
from ABDEGHJK pop D
from EGHJKAB pop H
from JKABEG pop A
from BEGJK pop G
from JKBE pop B
from EJK pop K
remain = EJ
```