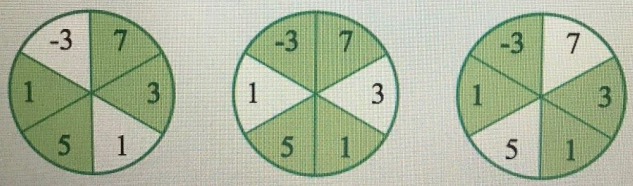
A farmer uses pivot irrigation to water a circular field of crops. Due to varying conditions, the field does not produce consistently. The farmer wants to achieve maximum profit using limited resources for harvest. The field is segmented into a number of equal segments, and a profit is calculated for each segment. This profit is the cost to harvest versus the sale price of the produce. The farmer will harvest a number of contiguous segments along with those opposite. Determine the maximum profit the farmer can achieve.

For example, the field is divided into $n = 6$ sections and will select $k = 2$ contiguous sections and those opposite for harvest. The profit estimates are *profit = [1, 5, 1, 3, 7, -3]* respectively. The diagrams below show the possible choices with *profits[0]* at the 9 o'clock position and filling counterclockwise.



The profit levels, from left to right, are *1 + 5 + 7 + 3 = 16*, *5 + 1 + 7 + -3 = 10*, and *1 + 3 + -3 + 1 = 2*. The maximum profit is *16*.

Function Description

Complete the function *maxProfit* in the editor below. The function must return the maximum profit achievable.

```
18
19  def maxProfit(k, profit):
20      halfLength = int(len(profit)/2)
21
22      newProfit = profit + profit[0:halfLength]
23      maxProfit = -10**9
24
25      for i in range(halfLength):
26          firstHalfProfit = sum(newProfit[i:i+k])
27          secondHalfProfit = sum(newProfit[i+halfLength:i+halfLength+k])
28          currProfit = firstHalfProfit + secondHalfProfit
29
30          if currProfit > maxProfit:
31              maxProfit = currProfit
32
33      return maxProfit
34
```