

题目描述：

“吃货”和“馋嘴”两人到披萨店点了一份铁盘（圆形）披萨，并嘱咐店员将披萨按放射状切成大小相同的偶数扇形小块。但是粗心服务员将披萨切成了每块大小都完全不同奇数块，且肉眼能分辨出大小。

由于两人都想吃到最多的披萨，他们商量了一个他们认为公平的分法：从“吃货”开始，轮流取披萨。除了第一块披萨可以任意选取以外，其他都必须从缺口开始选。

他俩选披萨的思路不同。“馋嘴”每次都会选最大块的披萨，而且“吃货”知道“馋嘴”的想法。

已知披萨小块的数量以及每块的大小，求“吃货”能分得的最大的披萨大小的总和。

输入描述：

第 1 行为一个正整数奇数 N ，表示披萨小块数量。 $3 \leq N < 500$ 。

接下来的第 2 行到第 $N+1$ 行（共 N 行），每行为一个正整数，表示第 i 块披萨的大小。 $1 \leq i \leq N$ 。披萨小块从某一块开始，按照一个方向依次顺序编号为 $1 \sim N$ 。每块披萨的大小范围为 $[1, 2147483647]$ 。

输出描述：

“吃货”能分得的最大的披萨大小的总和。

示例 1

输入：

5
8
2
10
5
7

输出：

19

说明：

此例子中，有 5 块披萨。每块大小依次为 8、2、10、5、7。按照如下顺序拿披萨，可以使“吃货”拿到最多披萨：

1、“吃货”拿大小为 10 的披萨

2、“馋嘴”拿大小为 5 的披萨

3、“吃货”拿大小为 7 的披萨

4、“馋嘴”拿大小为 8 的披萨

5、“吃货”拿大小为 2 的披萨

至此，披萨瓜分完毕，“吃货”拿到的披萨总大小为 $10+7+2=19$ 。

可能存在多种拿法，以上只是其中一种。

```
n = int(input())
a = [int(input()) for i in range(n)]
dp = [[-1] * n for i in range(n)]
def solve(L, R):
    if a[L] > a[R]:
        L = (L+1) % n
    else:
        R = (R+n-1) % n
    if dp[L][R] != -1:
        return dp[L][R]
    if L == R:
        dp[L][R] = a[L]
    else:
        dp[L][R] = max(a[L] + solve((L+1)%n, R), a[R] + solve(L, (R+n-1)%n))
    return dp[L][R]

ans = 0
for i in range(n):
    ans = max(ans, solve((i+1)%n, (i+n-1)%n) + a[i])

print(ans)
```