

题目描述：

Wonderland 是小王居住地一家很受欢迎的游乐园。*Wonderland* 目前有 4 种售票方式，分别为一日票（1 天）、三日票（3 天）、周票（7 天）和月票（30 天）。

每种售票方式的价格将由一个数组给出，每种票据在票面时限内可以无限制的进行游玩。例如，小王在第 10 日买了一张三日票，小王可以在第 10 日、第 11 日和第 12 日进行无限制的游玩。

小王计划在接下来一年内多次游玩该游乐园。小王计划的游玩日期将由一个数组给出。现在，请您根据给出的售票价格数组和小王计划游玩日期数组，返回完成游玩计划所需要的最低消费。

输入描述：

输入为 2 个数组

售票价格数组为 *costs*，*costs.length=4*，默认顺序为一日票、三日票、周票和月票。

小王计划游玩日期数组为 *days*， $1 \leq \text{days.length} \leq 365$ ， $1 \leq \text{days}[i] \leq 365$ ，默认顺序为升序。

输出描述：

完成游玩计划的最低消费

补充说明：

样例说明：

根据售票价格数组和游玩日期数组给出的信息，发现每次去玩的时候买一张一日票是最省钱的，所以小王会买 8 张一日票，每张 5 元，最低花费是 40 元。

示例 1

输入：

5 14 30 100

1 3 15 20 21 200 202 230

输出：

40

说明：

import sys

```
costs = list(map(int, input().strip().split()))
```

```
days = list(map(int, input().strip().split()))
```

```
plays = [1, 3, 7, 30]
```

```
dp = [costs[0] * i for i in range(len(days) + 1)]
```

```
for j in range(1, len(days) + 1):
```

```
    for i in range(1, len(costs) + 1):
```

```
        last_lay = days[j - 1] - plays[i - 1]
```

```
        last_lay_idx = 0
```

```
        for k in range(j - 1, -1, -1):
```

```
            if days[k - 1] <= last_lay:
```

```
                last_lay_idx = k
```

```
                break
```

```
        dp[j] = min(dp[j], dp[last_lay_idx] + costs[i - 1])
```

```
    #     print(dp, last_lay_idx)
```

```
    # print(dp, days[j - 1], "-")
```

```
print(dp[-1])
```