AcWing-4741：魔法百合井

森林里有一口很深的魔法井，井中有 L 朵百合花。

你带着一个大空篮子和足够多的硬币来到了井边。

这个井有魔力，向里面投入硬币可以发生神奇的事情：

- 如果你向井里一次性投入 1 个硬币，井就会发动魔法，将一朵百合花扔进你的篮子里。
- 如果你向井里一次性投入 4 个硬币，井就会发动魔法，统计并记录到目前为止，已经扔进你的篮子里的百合花的数量。
- 如果你向井里一次性投入 2 个硬币，井就会发动魔法，将等同于上次记录数量的百合花扔进你的篮子里。

有一点需要特别注意，如果你向井里一次性投入
1 个或
2 个硬币后，井中已经没有足够的百合花扔给你了，那么井就不会发动任何魔法，也不会扔给你任何百合花（钱白花了）。

请你计算，为了将所有百合花都收入篮中，所需要花费的最少硬币数量。

输入格式

第一行包含整数
T，表示共有
T 组测试数据。

每组数据占一行，包含一个整数 L，表示井中百合花的总数量。

输出格式

每组数据输出一个结果，每个结果占一行。

结果表示为 `Case #x: y`，其中
x 为组别编号（从
1 开始），
y 为需要花费的最少硬币数量。

数据范围

$1 \le T \le 100$,
$1 \le L \le 10^5$

难度：	中等
时/空限制：	1s / 64MB
总通过数：	1001
总尝试数：	1832
来源：	Google Kickstart2022 Round H Problem B
算法标签	

输入样例：

```
2
5
20
```

输出样例：

```
Case #1: 5
Case #2: 15
```

样例解释

对于 Case 1，井中一共有 5 朵百合花。

最佳方案是一个接一个的连续向井中投入
5 个硬币，这样我们可以一个接一个的得到
5 朵百合花。

一共需要花费 5 个硬币。

对于 Case 2，井中一共有 15 朵百合花。

最佳方案为：

- 首先，一个接一个的连续向井中投入
  5 个硬币，这样我们可以一个接一个的得到
  5 朵百合花。
- 然后，我们一次性向井中投入
  4 个硬币，这样井会记录下到目前为止扔进我们篮中的百合花数量为
  5。
- 最后，我们重复三次，每次向井中投入
  2 个硬币，这样每次都可以得到
  5 朵百合花，从而得到剩余的全部
  15 朵百合花。

一共需要花费 15 个硬币。

该题是个`dp`问题，子问题在于`dp[n]表示还剩n朵花时可以使用最少的硬币数将其取出`，那么可以划分子问题的求解方法：

- 实现的序列中最后一个投入是1
- 实现的序列中最后一个投入是42...2的序列

如果是第一个种情况，`dp[n]=min(dp[n],dp[n-1]+1)`

如果是第二种情况，需要枚举2的个数，枚举出所有情况中投币最少的情况。这种情况其实本质上可以理解为`1...142..2`的情况，那么前面1的个数就决定了这里4时获得数量，后续不断叠加的就是这个数量。因此需要满足整除关系，在枚举时可以直接枚举倍数值。

Code：

```
//
//  main.cpp
//  4741-魔法百何井
//
//  Created by MacBook Pro on 2023/7/19.
//

#include <iostream>
#include <cstring>
#include <algorithm>
using namespace std;
```

```cpp
const int N = 100005;    //计算所有数值后存储

int dp[N];               //剩余x朵花的最小银币数

int main() {
    memset(dp,0x3f,sizeof(dp));       //预处理为无穷大
    //计算所有值
    dp[0]=0;
    for(int i=1;i<N;i++){
        dp[i]=min(dp[i],dp[i-1]+1);      //考虑第一种情况
        for(int j=2;j*i<N;j++){
            //枚举倍数
            dp[i*j] = min(dp[i*j],dp[i]+4+2*(j-1));
        }
    }
    int T;
    cin>>T;
    for(int cases=1;cases<=T;cases++){
        int n;
        cin>>n;
        printf("Case #%d: %d\n",cases,dp[n]);
    }
    return 0;
}
```