

Split n into maximum composite numbers

将一个整数划分为合数之和——要求使用合数的数量最多

部分合数如下：

4 6 8 9 10 12 14 15 16 18 20...

对于任意数字n，（由于4是最小的合数，所以尽量使用4对n进行划分，可以得到最大的结果）：

1. 如果 $n < 4$ ，一定不可以划分
2. 如果 $n = 5, 7, 11$ ，一定不可以划分
3. 如果 $n \bmod 4 = 0$ ，那么直接返回 $n / 4$
4. 如果 $n \bmod 4 = 1$ ，令 $n - 9$ ，则新的n可以被4整除，所以结果是 $(n - 9) / 4 + 1$ （+1是因为9已经是一个合数了）
5. 如果 $n \bmod 4 = 2$ ，令 $n - 6$ ，则结果是 $(n - 6) / 4 + 1$
6. 如果 $n \bmod 4 = 3$ ，令 $n - 15$ ，则结果是 $(n - 15) / 4 + 2$ （+2 是因为 $15 = 9 + 6$ ）

```
#include <bits/stdc++.h>

using namespace std;

int splitN(int &N) {
    if (N < 4) return -1;
    int remainder = N % 4;
    if (remainder == 0) return N / 4;
    if (remainder == 1) {
        // eg 5
        if (N < 9) {
            return -1;
        }
        return (N - 9) / 4 + 1;
    } else if (remainder == 2) {
        return (N - 6) / 4 + 1;
    } else if (remainder == 3) {
        // eg 7 11
        if (N < 15) return -1;
        // 15 = 6 + 9
        else return (N - 15) / 4 + 2;
    }
}
```

```
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        printf("%d\n", splitN(N));
    }
}
```