

LCM Permutation

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Construct a permutation $p_1, p_2, \dots, p_n$ of length n that satisfies $\text{lcm}_{1 \leq i \leq n} (p_i + p_{(i \bmod n)+1}) \leq 20n$.

Input

The first line contains a positive integer t ($1 \leq t \leq 10^4$), representing the number of test cases. Each test case contains a single integer n ($1 \leq n \leq 8 \cdot 10^6$), representing the length of the permutation. It is guaranteed that the sum of n over all test cases does not exceed $8 \cdot 10^6$.

Output

For each test case, output the permutation $p_1, p_2, \dots, p_n$ that satisfies the condition.

It is guaranteed that at least one such permutation exists.

Example

standard input	standard output
1 3	1 2 3

Note

In the example, $\text{lcm}(p_1 + p_2, p_2 + p_3, p_3 + p_1) = \text{lcm}(3, 5, 4) = 60 \leq 20n$.

You can use the following code to speed up the output.

```
#include <bits/stdc++.h>
using namespace std;

int ptr = 0; char buf[1 << 23];
inline void writeChar(char c) {
    if (ptr == 1 << 23) fwrite(buf, 1, ptr, stdout), ptr = 0; buf[ptr++] = c;
}
inline void writeInt(int x) {
    int t = 0; char s[10];
    while (x) s[t++] = (char)('0' + x % 10), x /= 10;
    while (t) writeChar(s[--t]);
}
inline void flushOut() {
    if (ptr) fwrite(buf, 1, ptr, stdout), ptr = 0;
}

int main() {
    // Your code here...
    flushOut();
    return 0;
}
```