Problem Statement

Santa has n candies and he wants to gift them to k kids. He wants to divide as many candies as possible between all k kids. Santa can't divide one candy into parts but he is allowed to not use some candies at all.

Santa will be satisfied if the following conditions are met:

1. The difference between the kid who receives the maximum (b) and the minimum (a) number of candies is at most 1, i.e., $b - a \leq 1$.
2. The number of kids who receive $a + 1$ candies does not exceed $\lfloor k/2 \rfloor$ (where $\lfloor k/2 \rfloor$ is k divided by 2 and rounded down).

Your task is to find the maximum number of candies Santa can distribute while satisfying the conditions.

Input

- The first line contains a single integer t ($1 \leq t \leq 50{,}000$) — the number of test cases.
- Each of the next t lines contains two integers n and k ($1 \leq n, k \leq 10^9$) — the number of candies and the number of kids.

Problem Analysis

Santa needs to distribute n candies to k kids such that:

1. The difference between the maximum (b) and minimum (a) candies given to any kid is at most 1.
2. The number of kids receiving $a + 1$ candies does not exceed $\lfloor k/2 \rfloor$.

Key Observations

- **Base Distribution**: Each kid can receive $a = n \,//\, k$ candies as a baseline.
- **Remainder Handling**: After distributing a candies to all kids, there may be a remainder $r = n \,\%\, k$ candies left.
- **Constraint Compliance**: Only $x = \lfloor k/2 \rfloor$ kids can receive the extra $a + 1$ candies.

Algorithm

1. **Compute x**:
   $x = \lfloor k \,/\, 2 \rfloor$
   This represents the maximum number of kids allowed to receive $a + 1$ candies.

2. **Calculate Base Distribution**:

   ```
   \text{base} = (n // k) \times k
   ```

   This is the total candies distributed if all kids get a candies.

3. **Compute Remainder**:

```
r = n - \text{base}
```

This represents the leftover candies after the base distribution.

4. **Adjust for Constraints**:
   - **If** $r \leq x$: Distribute all r candies to r kids as $a + 1$.
   - **If** $r > x$: Distribute x candies to x kids as $a + 1$, leaving $r - x$ candies unused.

By following this approach, we maximize the number of candies distributed while ensuring Santa's conditions are met.

Example Walkthrough

Test Case

Given:
$n = 11$, $k = 3$

Step-by-Step Calculation

1. **Compute** x:
   [
   x = \lfloor 3 / 2 \rfloor = 1
   ]
   This means at most 1 kid can receive $a + 1$ candies.

2. **Calculate Base Distribution**:
   [
   \text{base} = (11 // 3) \times 3 = 3 \times 3 = 9
   ]
   Each kid gets at least $a = 3$ candies.

3. **Compute Remainder**:
   [
   r = 11 - 9 = 2
   ]
   2 candies are left to be distributed.

4. **Adjust for Constraints**:
   [
   \text{total} = 9 + \min(2,1) = 9 + 1 = 10
   ]
   Since $r > x$, only $x = 1$ extra candy can be given.

Distribution

- 2 kids receive 3 candies (a).

- `1` kid receives `4` candies (`a + 1`).

Total Distributed:
[
3 + 3 + 4 = 10
]
which satisfies both constraints.

Code

```
import sys

def solve():
    n, k = map(int, sys.stdin.readline().split())
    x = k // 2
    total = (n // k) * k
    total += min(n - total, x)
    print(total)

def main():
    sys.stdin.readline()
    for _ in range(int(sys.stdin.readline().strip())):
        solve()

if __name__ == "__main__":
    main()
```