❖ Problem Link : https://codeforces.com/problemset/problem/1475/B

❖ Introduction

Polycarp loves the years **2020** and **2021** , and he wants to express a number **n** as a sum using only these two values.
Your task is to determine whether such a representation is possible for each test case.

❖ Problem Explanation

Given a number **n**, we want to check whether :

n = (2020 × a) + (2021 × b)

for some non-negative integers a , b

➢ Examples :

- 4041 = 2020 + 2021 -> YES
- 4042 = 2021 + 2021 -> YES
- 8079 cannot be formed -> NO

❖ Full Python Code :

```
t = int(input())
for _ in range(t):
  n = int(input())
  y = n // 2020
  x = n % 2020
  if x <= y:
    print("YES")
  else:
    print("NO")
```

❖ Algorithm Explanation

Key idea :

➢ Observation:

We want to check if a number **n** can be formed using:

- 2020 × a
- 2021 × b

where a and b are non-negative integers.

So we want :

N = 2020 a + 2021 b

Notice that :

2021 = 2020 + 1

So a 2021- value is just a 2020-value plus 1.

Simple Calculation :

Try to use as many 2021's as possible because each 2021 contributes one extra on top of 2020.

Let's rewrite n like this:

n = 2020 a + (2020 + 1) b = 2020 (a + b) + b

This means:

n is possible if and only if the remainder when dividing by 2020 is ≤ the number of 2021s you use.

Simpler explanation for beginners :

- Buying a 2021 - package is like buying a 2020-package plus 1 extra.
- So if you use b copies of 2021, you get b extra units.
- Therefore, you only need to check:

Does n leave a remainder a ≤ b when divided by 2020?

Example (easy steps) :

n = 4041

Try b = 1 (one 2021) :

4041 − 2021 = 2020

2020 is divisible by 2020 →YES

---

n = 4042

Try b = 2 :

4042 – 2 × 2021 = 0

0 is divisible → YES

---

n = 8079

Try b = 1 :

8079 – 2021 = 6058 (not divisible by 2020)

Try b = 2 :

8079 – 4042 = 4037 (not divisible)

Try b = 3 :

8079 – 6063 = 2016 (not divisible)

No value works → NO

Thus :

- Let **x = n % 2020** (the remainder when dividing by 2020) .
- Let **y = n // 2020** .

Then representation is possible **if and only if** : x ≤ y

Because every time you replace one 2020 by 2021, the total increases by 1.

2021 = 2020 + 1,
each time you use a 2021 instead of a 2020, the total increases by +1.

So 2021s let you add extra 1s on top of a combination of 2020s.

Here :

(/) → represents normal division which gives (int + float) output .

(/ /) → represents Floor division which gives (int) output .

(%) → represents modulo which gives remainder of a division .

❖ Step-by-Step Working

Input:

5
1
4041
4042
8081
8079

Working:

1. **n = 1**
   - n // 2020 = 0
   - n % 2020 = 1
   - 1 ≤ 0 -> NO
2. **n = 4041**
   - 4041 // 2020 = 2
   - 4041 % 2020 = 1
   - 1 ≤ 2 -> YES
3. **n = 4042**
   - 4042 // 2020 = 2
   - 4042 % 2020 = 2
   - 2 ≤ 2 -> YES
4. **n = 8081**
   - 8081 // 2020 = 4
   - 8081 % 2020 = 1
   - 1 ≤ 4 -> YES
5. **n = 8079**
   - 8079 // 2020 = 4
   - 8079 % 2020 = - actually 8079 % 2020 = ??
     2020 × 3 = 6060
     2020 × 4 = 8080 -> too big
     So 8079 % 2020 = 8079 - 6060 = 2019
   - 2019 ≤ 4 -> NO

Output

```
NO

YES

YES

YES

NO
```

❖ Time & Space Complexity

Operations	Complexity
Per test calculation	**O (1)**
For t test cases	**O (t)**
Space usage	**O (1)**
Total Time Complexity	**O (t)**

❖ Conclusion

The key insight is that the remainder when dividing **n** by 2020 must be less than or equal to the quotient. This mathematical trick removes the need for loops and makes the solution efficient for up to 10^4 test cases.

One sentence result

A number n can be written as 2020·a + 2021·b
if the remainder when dividing by 2020 is not larger than the quotient .

Thank You