1. Pick out the correct output :

```
main() {
static int var = 5;
printf("%d ",var--);
if(var)
main();
}
```

a) 5 4 3 2 1 b) 5
c) Compilation error d) infinite loop

2. The output is :

```
int main() {
extern out=10;
out*=10;
printf(“%d”,out);
}
```

a) 0 b) 10
c) 100 d) Compilation error

3. What will be the output of the following code?

```
#include <stdio.h>
int main() {
int a[][3] = { 1,2,3 ,4,5,6};
int (*ptr)[3] =a;
printf("%d %d " ,(*ptr)[1], (*ptr)[2] );
++ptr;
printf("%d %d" ,(*ptr)[1], (*ptr)[2] );
return 0;
}
```

a) 2 3 5 6 b)1 1 2 2
c) 1 1 4 4 d)garbage value

4. What will the printf statement in the following code print?

```
main ( ) {
static char *s*+=,“black”,“white”,“yellow”,“violet”-;
char **ptr[] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
**++p;
printf(“%s”,*--*++p + 3);
}
```

a) white b) yellow
c)ck d) No display

5. Practical numbers have for long fascinated man. In mathematics, and in particular number theory, a practical number is a positive integer n such that all smaller positive integers can be represented as sums of distinct divisors of n.
For example, 12 is a practical number because all the numbers from 1 to 11 can be expressed as sums of its divisors 1, 2, 3, 4, and 6: as well as these divisors themselves, we have 5=3+2, 7=6+1, 8=6+2, 9=6+3, 10=6+3+1, and 11=6+3+2.
Some practical numbers are 1, 2, 4, 6, 8, 12, 16, 18, 20, 24, 28...
Great mathematicians like Fibonacci and Srinivasan loved dabbling with Practical numbers. It is your turn now!!
Write a program that tests whether a number is practical or not.

INPUT

The first line of the input contains an integer N(1<=N<=10). Next N lines contain an integer K each (1<=K<=100000).

OUTPUT

For each test case print 'Yes' if the number is practical and 'No' otherwise.

Example case

Input

10
24
47
88
356
4096
7864
8700
9198
9804
9999

Output

Yes
No
Yes
No
Yes
No
Yes
Yes
Yes
No