C. Divisibility by Eight

time limit per test

2 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

You are given a non-negative integer *n*, its decimal representation consists of at most 100 digits and doesn't contain leading zeroes.

Your task is to determine if it is possible in this case to remove some of the digits (possibly not remove any digit at all) so that the result contains at least one digit, forms a non-negative integer, doesn't have leading zeroes and is divisible by 8. After the removing, it is forbidden to rearrange the digits.

If a solution exists, you should print it.

**Input**

The single line of the input contains a non-negative integer *n*. The representation of number *n*doesn't contain any leading zeroes and its length doesn't exceed 100 digits.

**Output**

Print "NO" (without quotes), if there is no such way to remove some digits from number *n*.

Otherwise, print "YES" in the first line and the resulting number after removing digits from number *n* in the second line. The printed number must be divisible by 8.

If there are multiple possible answers, you may print any of them.

**Examples**

**input**

3454

**output**

YES  
344

**input**

10

**output**

YES  
0

**input**

111111

**output**

NO

#include<iostream>

#include<vector>

#define MAX 1000

using namespace std;

vector<pair<int,int>>store;

vector<int>food;

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

using namespace std;

char a[222];

int main() {

scanf("%s", a + 3);

a[1] = a[2] = '0';

for (int i = 1; a[i]>0; i++) {

for (int j = i + 1; a[j]>0; j++) {

for (int k = j + 1; a[k]>0; k++) {

int w = (100 \* (a[i] - '0') + 10 \* (a[j] - '0') + a[k] - '0');

if (w % 8 == 0)

{

puts("YES"); printf("%d\n", w);

return 0;

}

}

}

}

puts("NO");

}

代码的意思就是先在字符串里加两个先导0,如此一来后面进行的暴搜就可以先一个位一个位搜然后在两个位两个位搜最后三个位三个位搜，这个加先导零的方法非常优雅，至于为什么只搜三个位，这就是这道题的精华之处了。

先说说这道题的推理过程：

大部分人看到这道题，第一个想法是怎么样暴力搜索（事实上这道题是可以暴搜过的，只要从小往大了搜的话一定过，原因见后），比较二的比如我想的是这个求余是不是可以玩出花来。

但是这道题正确的解法是，先去思考能被8整除的数字的范围（啊和我的想法只差一点点距离）然后就发现 1000是能被8整除的，也就是说，所有超过三位数的数字，都可以只看百位到个位能否被8整除（因为千位和更大的位数都能被8整除），然后发现那么只要找到一个三位数可以被8整除的数字就好了。

推理真有意思hhh