【题目大意】

给一个不超过100位的数字，要求能否删掉几位数，剩下的数能被8整除

【解题思路】：这里有个性质：如果一个数后三位能被8整除，那么这个数就能被8整除

换句话说，如果一个大于三位的数能被8整除，那么其后三位一定能被8整除

因此，只需要枚举1位数，2位数，3位数就行了

大于三位数能被8整除的，一定会在3位数里面被枚举出来（他的后三位首先一定会被8整除）

证明：举一个5位数的例子吧，

例如

\_\_\_\_\_ \_\_\_\_\_ \_\_ \_\_ \_\_ \_\_ \_\_\_

abcde=ab000+cde=1000×ab+cde=8×125×ab+cde

很明显，8×125×ab一定是8或者125的倍数，因此当cde能被8或者125整除时，五位数abcde就能被8或者125整除。位数再多也是一样的，主要是1000=125\*8

那么只要枚举后三位即可

注意在枚举一位的时候有个坑点，不只是8, 0也可以被8整除

#include<stdio.h>

#include<string.h>

#include<iostream>

#include<set>

#include<algorithm>

#include<map>;

typedef long long ll;

using namespace std;

int main()

{

// freopen("input.txt","r",stdin);

string s;

cin>>s;

int l=s.size();

for(int i=0;i<l;i++)//枚举一位的

if((s[i]-'0')==8||(s[i]-'0')==0)

{

puts("YES");

cout<<s[i]-'0'<<"\n";

return 0;

}

for(int i=0;i<l;i++)//枚举两位的

for(int j=i+1;j<l;j++)

if(((s[i]-'0')\*10+(s[j]-'0'))%8==0)

{

puts("YES");

cout<<(s[i]-'0')\*10+(s[j]-'0')<<"\n";

return 0;

}

for(int i=0;i<l;i++)//枚举三位的

for(int j=i+1;j<l;j++)

for(int k=j+1;k<l;k++)

if(((s[i]-'0')\*100+(s[j]-'0')\*10+(s[k]-'0'))%8==0)

{

puts("YES");

cout<<(s[i]-'0')\*100+(s[j]-'0')\*10+(s[k]-'0')<<"\n";

return 0;

}

puts("NO");

return 0;

}