

Bottom Coder (Hard)

Some of you may be familiar with the TopCoder (TM) contest. Its exact rules are not important for this problem, but know that the most important part of it is writing a program according to the given specification. Many times the contestant ends up with a program which would work perfectly – if only he could change a couple of characters (like, replacing "=" by "==" in C, etc.). Unfortunately, even the best programmers sometimes aren't able to spot these tiny but necessary changes until it's too late... and that's why we developed a brand-new BottomCoder training for them!

The idea is very simple – you're given a problem specification, a source code, and a list of permitted modifications. Your task is to find a modification which would cause the program to behave according to the specification.

Specification: "Write a program which outputs a short English text mentioning our partner competition – IPSC. The text must consist of one or more English sentences and each sentence has to contain one or more English words (sequences of only upper-case characters) separated by spaces. Additionally, you may use certain punctuation characters – namely ".!?,\". Try to obfuscate the program as much as possible." The code you are supposed to modify:

```
#include <stdio.h>

int rex[5];

void f3(int *a) {
    int i;
    for (i=0; i<5; i++) a[i]=0;
}

int f2(int *a) {
    int i;
    for (i=0; i<5; i++) if (a[i]!=0) return 0;
    return 1;
}

void f1(int *a) {
    int i;
    for (i=0; i<5; i++) {
        a[i]++;
        if (a[i]<100) break;
        a[i]-=100;
    }
    for (i=4; i>=0 && a[i]>=rex[i]; i--)
        if (a[i]>rex[i])
            f3(a);
}

void f4(int *a) {
    int i;
    for (i=0; i<5; i++) {
        a[i]--;
        if (a[i]>=0) break;
        a[i]+=100;
    }
}
```

```

    if (i>=5) for (i=0; i<5; i++) a[i]=rex[i];
}

void f7(int *a, int *b) {
    int c[5];
    f3(c); f3(a);
    while(!f2(b)) { f1(a); f4(b); f1(c); }
    while(!f2(c)) { f1(b); f4(c); }
}

void f9(int *a, int *b) {
    f1(a);
    while(!f2(b)) { f4(b); f1(a); }
}

void f8(int *a, int *b) {
    int c[5], d[5];
    f7(d, a);
    f3(a); f1(a);
    while(!f2(b)) { f7(c, d); f9(a, c); f4(a); f4(b); }
}

void f5(int *a, int *b) {
    int c[5], d[5];
    f7(d, a);
    f3(a); f1(a);
    while(!f2(b)) { f7(c, d); f8(a, c); f4(a); f4(b); }
}

void f10(int x) {
    int rpl[]=
{80, 125, 111, 18, 59, 88, 88, 28, 65, 98, 119, 103, 101, 79, 107, 2, 16,
92, 102, 123, 103, 84, 112, 78, 68, 98, 65, 37, 105, 85, 107, 13, 45, 9,
104, 81, 21, 31, 55, 110, 78, 66, 66, 3, 77, 63, 16, 105, 15, 123, 16, 84,
31, 96, 4, 82, 82, 122, 68, 115, 35, 73, 3, 108, 115, 83, 15, 19, 31, 99, 5,
123, 24, 65, 36, 15, 75, 84, 4, 2, -1};

    int i;
    int a[5], b[5], c[5];

    if (x<100000000 || x>200000000) return;
    x--;
    f3(rex); rex[4]=1;
    for (i=0; rpl[i]!=-1; i++)
    {
        f3(a); a[0]=i+1;
        f3(b); f1(b); f3(c); f1(b); f1(b);
        f1(c); f1(b); f5(a, b);

        f1(c);
        while(!f2(a))
        {
            f3(b); b[0]=x%100; b[1]=x/100;
            f4(a); f8(c, b);
        }
        rpl[i]^=c[1];
        printf("%c", rpl[i]);
    }
    printf("\n");
}

```

```
int main()
{
    f10(47);
}
```

As you can see, the coder made almost everything according to the specification :) You're only allowed to alter one number in the source code – namely the number 47 on line 98 (the argument of function "f10" called from "main"). You can replace it by any integer between 100`000`000 and 200`000`000 inclusive.

Input

There is no input for given problem.

Output

Your submission should consist of two lines. The first line should contain the value of the constant – an integer between 100`000`000 and 200`000`000 inclusive. The second line should contain the output produced by the program if it were compiled and executed with the correct value of the constant.

Example

Output:

123456789

ARE YOU SOLVING IPSC PROBLEMS RIGHT NOW?

(syntactically valid (but incorrect) submission)