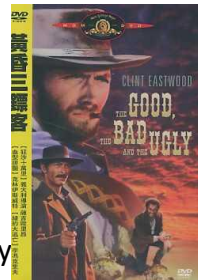


Homework #4

Due date: 12/1

The good, the bad and the ugly



Given an unsigned integer ≥ 1 , determine if it is good, bad, and/or ugly

Good number

An unsigned integer is **good** if all the numbers formed by 3 adjacent digits of it are distinct. That is, the number $d_k d_{k-1} d_{k-2} \cdots d_2 d_1 d_0$ is good if the numbers

$$d_k d_{k-1} d_{k-2}, d_{k-1} d_{k-2} d_{k-3}, \cdots, d_2 d_1 d_0$$

are all distinct.

Note that an unsigned integer is good if it is formed by 3 digits or less.

For examples, the following numbers are good

1

1234567890 since 123, 234, 345, 456, 567, 678, 789, 890 are all distinct

and the following numbers are not

1093500000 since 000 occurs thrice

3456565657 since 565 occurs thrice and 656 occurs twice

Required algorithm

Your algorithm shall extract all the 3-digit numbers from the given unsigned integer, store them in an array, and compare them *pairwise* for inequality.

The array shall not be oversized – it shall be of the type

short[8]

or

unsigned short[8]

since a 4-byte unsigned integer has at most 10 decimal digits (so that there are at most 8 3-digit numbers) and the maximum of a 2-byte (unsigned) short integer exceeds every 3-digit number.

Bad number

An unsigned integer is **bad** if it contains the digits 1, 3, and 5.

Note that a bad unsigned integer must have at least 3 digits.

For examples, the following numbers are bad

135135

1234567890



and the following numbers are not

1 since 3 and 5 don't occur in it

3456565657 since 1 doesn't occur in it

Required algorithm

Your algorithm shall count the number of times each decimal digit occurs in the given unsigned integer. To this end, you shall declare an array, say

char d[10]={0}; // initialize $d[k] = 0, 0 \leq k \leq 9$

or

unsigned char d[10]={0}; // initialize $d[k] = 0, 0 \leq k \leq 9$

and then, for all $k, 0 \leq k \leq 9$, set

$d[k] =$ the number of times digit k occurs in the unsigned integer

Notice that a char is used as a tiny integer here. Also, either signed or unsigned char will do, because a decimal digit can occur at most 10 times in the unsigned integer.

Finally, the unsigned integer is bad iff $d[1] \geq 1, d[3] \geq 1$ and $d[5] \geq 1$.

Example

For 3456565657, we have

index	0	1	2	3	4	5	6	7	8	9
d	0	0	0	1	1	4	3	1	0	0

This number is not bad, since $d[1] \not\geq 1$.

Ugly number

An unsigned integer ≥ 1 is **ugly** if its only prime factors are 2, 3 or 5.

By convention, 1 is an ugly number.

For examples, the following numbers are ugly

1937102445 since $1937102445 = 3^{18}5^1$

2361960000 since $2361960000 = 2^63^{10}5^4$

and the following numbers are not

135135 since $135135 = 3^35^17^111^113^1$

1234567890 since $1234567890 = 2^13^25^13607^13803^1$

Requirements

- 1 Your program shall contain the following three functions.

bool good(unsigned n); // determine if n is good

bool bad(unsigned n); // determine if n is bad

bool ugly(unsigned n); // determine if n is ugly

- 2 Use the stipulated algorithms, if any.
- 3 Properly comment your program
- 4 Refer to the sample run for the required output format.

Sample run

```
Enter an unsigned integer >= 1: 1937102445
Good, Bad, Ugly
```

```
Enter an unsigned integer >= 1: 1234567890
Good, Bad, Not ugly
```

```
Enter an unsigned integer >= 1: 1
Good, Not bad, Ugly
```

```
Enter an unsigned integer >= 1: 987643210
Good, Not bad, Not ugly
```

```
Enter an unsigned integer >= 1: 1093500000
Not good, Bad, Ugly
```

```
Enter an unsigned integer >= 1: 135135
Not good, Bad, Not ugly
```

```
Enter an unsigned integer >= 1: 2361960000
Not good, Not bad, Ugly
```

```
Enter an unsigned integer >= 1: 3456565657
Not good, Not bad, Not ugly
```

```
Enter an unsigned integer >= 1: ^Z
```