Homework #4

Due date: 11/4

Part A: The good, the bad and the ugly (100%)

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

Good number

An unsigned integer $d_k \cdots d_1 d_0$ is good if

$$\sum_{i=0}^{k} \sum_{j=i}^{k} d_j = 'g' + 'o' + 'o' + 'd'$$

Note that the sum of the ASII codes of $'g','o','o'$, and $'d'$ is equal to 425.

Example

2345 not good, $\because 2 + (2+3) + (2+3+4) + (2+3+4+5) = 30$

2999999999 good,

$\because 2 + (2+9) + (2+9+9) + (2+9+9+9) + \cdots + (2+9+9+\cdots+9) = 425$

Required algorithm

Since a 4-byte unsigned integer has at most 10 decimal digits, you shall declare

`char d[10];` // either signed or unsigned will do (why?)

Step 1 Extract the digits and set `d[`i`]`$= d_i,\ 0 \leq i \leq k$

Step 2 Compute `d[`i`]`$= \sum_{j=i}^{k} d_j,\ 0 \leq i \leq k$ incrementally, i.e. use `d[`i`]`$= \sum_{j=i}^{k} d_j$ to compute `d[`$i-1$`]`$= \sum_{j=i-1}^{k} d_j$

Step 3 Sum up the array elements, yielding the value $\sum_{i=0}^{k} \sum_{j=i}^{k} d_j$

Example – For 2345, we have

Step 1:

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

Step 2:

index	9	8	7	6	5	4	3	2	1	0
d							2	5	9	14

Step 3: Compute $2 + 5 + 9 + 14 = 30$

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.

Example

503301	bad
1234567890	bad
1	not bad, $\because$ 3 and 5 don't occur in it
3456565657	not bad, $\because$ 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 first declare an array, say

```
char d[10]={0};               //* either signed or unsigned will do (why?)
```

that initializes $\mathbf{d}[i] = 0,\ 0 \le i \le 9$.
Next, for all i, $0 \le i \le 9$, set
$\mathbf{d}[i]$ = the number of times digit i occurs in the unsigned integer.
It follows that the unsigned integer is bad iff $\mathbf{d}[1] \ge 1$, $\mathbf{d}[3] \ge 1$ and $\mathbf{d}[5] \ge 1$.

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 $\mathbf{d}[1] \ngeq 1$.

Ugly number

An unsigned integer $n \ge 1$ is ugly if its only prime factors are 2, 3 *or* 5. In other words, $n = 2^i 3^j 5^k$, for some $i, j, k \ge 0$.

Example

1937102445	ugly, $\because$ $1937102445 = 3^{18}5^{1}$
2361960000	ugly, $\because$ $2361960000 = 2^{6}3^{10}5^{4}$
135135	not ugly, $\because$ $135135 = 3^{3}5^{1}7^{1}11^{1}13^{1}$
1234567890	not ugly, $\because$ $1234567890 = 2^{1}3^{2}5^{1}3607^{1}3803^{1}$

Required algorithm

Factor out 2, 3 and 5 and check whether the remaining is 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

3 Properly comment your program

4 Refer to the sample run for the required output format.

Sample run

```
Enter an unsigned integer >= 1: 3999998888
Good, Not bad, Not ugly

Enter an unsigned integer >= 1: 1937102445
Not good, Bad, Ugly

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

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

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

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

Part B: Ugly number generator (100%)

Ugly numbers are also known as Hamming numbers or 5-smooth numbers.
The Hamming sequence is an ascending sequence of Hamming numbers.
For example, the first 30 Hamming numbers are 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54, 60, 64, 72, 75, and 80.

Consider the following two problems:

1. Recognition problem
   Given an unsigned integer, determine if it is an ugly number.
2. Generation problem
   Given an integer $n \geq 0$, generate the n^{th} ugly number.

The recognition problem can be easily solved. The previous function

`bool ugly(unsigned n);`

is a recognizer that solves this problem.

The generation problem is much harder than the recognition problem. For this part, you are asked to write the following generator:

`unsigned ugly(unsigned n);` // generate the nth ugly number

Required algorithm

The following algorithm is based on the formal definition of Hamming sequence H:

1. $1 \in H$
2. If $x \in H$, then $2x, 3x, 5x \in H$
3. Nothing else is in H

Note that $1 \in H$ is the 0^{th} Hamming number.

Let $H[n+1]$ be a semidynamic array.
In the sequel, the notation

$$H = \left\{ a, \underbrace{b}_{3x,5x}, \underbrace{c}_{2x}, d, e \right\}$$

means the first five Hamming numbers
$H[0] = a, H[1] = b, H[2] = c, H[3] = d, H[4] = e$
have already been generated and the Hamming number $H[5]$ to be generated next is the minimum of $2c, 3b$ and $5b$.

The algorithm proceeds as follows:

Initialize $H = \left\{ \underbrace{1}_{2x,3x,5x} \right\}$

$$\Rightarrow H = \left\{ \underbrace{1}_{3x,5x}, \underbrace{2}_{2x} \right\} \qquad \because \min\{2,3,5\} = 2$$

$$\Rightarrow H = \left\{ \underbrace{1}_{5x}, \underbrace{2}_{2x,3x}, 3 \right\} \qquad \because \min\{4,3,5\} = 3$$

$$\Rightarrow H = \left\{ \underbrace{1}_{5x}, \underbrace{2}_{3x}, \underbrace{3}_{2x}, 4 \right\} \qquad \because \min\{4,6,5\} = 4$$

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{3x,5x}, \underbrace{3}_{2x}, 4,5 \right\} \qquad \because \min\{6,6,5\} = 5$$

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{3x,5x}, 3, \underbrace{4}_{2x}, 5,6 \right\} \qquad \because \min\{6,6,10\} = 6 \text{ (Select } 2x3\text{. See below)}$$

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{5x}, \underbrace{3}_{3x}, \underbrace{4}_{2x}, 5,6 \right\} \qquad \because \min\{8,6,10\} = 6 \text{ has already been found}$$

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{5x}, \underbrace{3}_{3x}, 4, \underbrace{5}_{2x}, 6,8 \right\} \qquad \because \min\{8,9,10\} = 8$$

$\Rightarrow$ *and so on*

Comment

In this step, it is equally well to select $3x2$, as it will eventually yield the same result:

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{5x}, \underbrace{3}_{2x,3x}, 4,5,6 \right\} \qquad \because \min\{6,6,10\} = 6 \text{ (Select } 3x2\text{)}$$

$$\Rightarrow H = \left\{ 1, \underbrace{2}_{5x}, \underbrace{3}_{3x}, \underbrace{4}_{2x}, 5,6 \right\} \qquad \because \min\{6,9,10\} = 6 \text{ has already been found}$$

Suggestion

It is up to you to decide how to implement the algorithm. However, it is suggested that the following arrays be used:

```
unsigned H[n+1];
const unsigned x[3]={2,3,5};    // 2x, 3x, and 5x
int index[3];
unsigned ugly[3];               // the next 3 ugly numbers
```

The relationship among these four arrays is characterized by the equation:

```
ugly[i]=x[i]*H[index[i]]
```

Hint

Q: How to find the minimum of three numbers **a**, **b** and **c**?

A:
```
min=a;
if (b<min) min=b;
if (c<min) min=c;
```

Sample run

```
Enter an unsigned integer: 0
H[0] = 1

Enter an unsigned integer: 99
H[99] = 1536

Enter an unsigned integer: 500
H[500] = 944784

Enter an unsigned integer: 1000
H[1000] = 51840000

Enter an unsigned integer: ^Z
```