

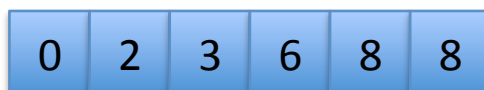
ACM ICPC 2012 [国内予選問題 B](#) 「繰り返す 10 進数」

● 数列の次の値を求める方法

1. 配列  $d[i]$  に 10 進数の第  $i$  桁目の数値をいれる ( $0 \leq i < L$ )。<sup>1</sup>
2. 配列  $d[]$  を昇順にソートする<sup>2</sup>。  $d[0] \leq d[1] \leq \dots \leq d[L-1]$
3. ソート結果より、最大値  $\max$  と 最小値  $\min$  を求める。<sup>3</sup>
4.  $\max - \min$  が求める値である。



桁を大きさの順にソート



上位            . . . .            下位    → 最小値 = 886320  
最大値 = 886320 ← 下位            . . . .            上位

●  $a_i = a_j$  (ただし  $i > j$ ) となる最小の  $i$  を求める方法

- ◆  $a_i$  が得られたら、 $a[i] = a_i$  とし、 $a[0] \sim a[i-1]$  の中に  $a[i] = a[j]$  となる  $a[j]$  が存在するか調べる。
- ◆ そのような  $a[j]$  が見つければ、その時の  $i, j$  が求める  $i, j$  である。
- ◆ 見つからなければ、 $i = i + 1$  として繰り返す。
- ◆ なお、求める  $i$  は 20 を超えないと仮定して構わないので、配列  $a$  の大きさは 21 あれば良い。

<sup>1</sup> 10 進数  $x$  に対し、 $x \% 10$  は最下位桁 (下から 0 桁目) の値を表す。次に、 $x = x/10$  とすると  $x \% 10$  は元の  $x$  の下から 1 桁目の数値となる。これを繰り返すことにより、1 桁ずつ値を求めることができる。

<sup>2</sup>  $L$  は 6 以下なので、バブルソートなどの単純なアルゴリズムで良い。

<sup>3</sup>  $d[0]$  を最下位桁、 $d[L-1]$  を最上位桁と見なすと最大値  $\max$  となり、逆に、 $d[0]$  を最上位桁、 $d[L-1]$  を最下位桁と見なすと最小値  $\min$  となる。 $\max$  や  $\min$  を求めるには、最初  $\max = \min = 0$  としておき、最上位桁から順に、 $\max$  と  $\min$  を 10 倍してその桁の値を加算する操作を最下位桁まで繰り返せば良い。

### 【プログラム例 1】

```
// ICPC 2012 国内予選問題 B
//
// Filename = pb1.c
// Compile : cc pb1.c
// Execution : ./a.out < B0 > B0.result
// Check : diff B0.ans B0.result
//
#include <stdio.h>

#define MAX_DIGITS 6
#define MAX_I 20

// 数列の x の次の値を計算して返す。el は桁数。
int next_val(int x, int el)
{
    int d[el]; // 10 進数の各桁の値を入れる配列
    int i, j, tmp;
    int max, min;
    // まず 10 進数 x の i 桁目の値を d[i] にセットする
    for(i=0; i<el; i++){
        d[i] = x % 10; // 元の x の下から i 桁目の値が d[i] に格納される
        x = x/10;
    }
    // 次に配列 d を昇順にソートする。(バブルソート)
    for(j=el-2; j>=0; j--){
        for(i=0; i<=j; i++){
            if(d[i] > d[i+1]){
                tmp = d[i];
                d[i] = d[i+1];
                d[i+1] = tmp;
            }
        }
    }
    // 最大値と最小値を求める
    max = min = 0;
    for(i=0; i<el; i++){
        min = min*10 + d[i];
        max = max*10 + d[el-1-i];
    }
    return max - min; // 数列の次の値を返す
}

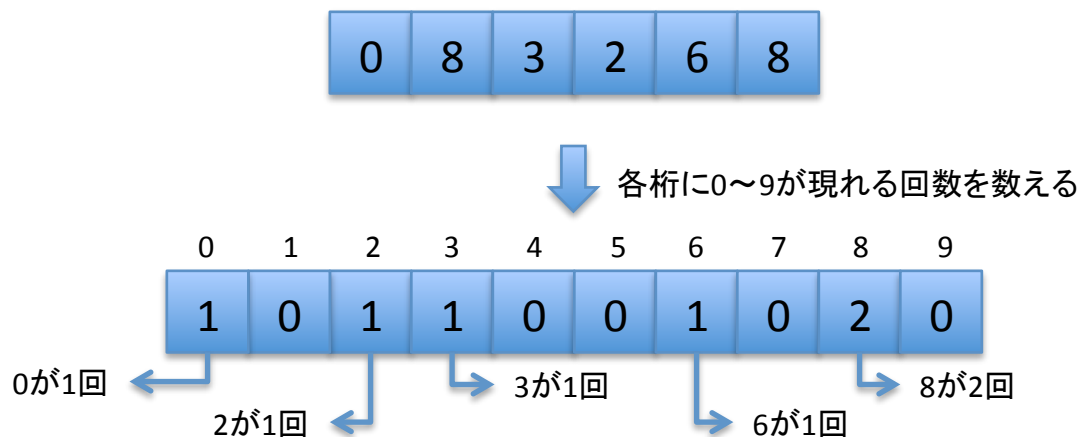
int main()
{
    int a[MAX_I + 1]; // 数列の値を格納する配列
    int el; // 桁数を格納する変数
    int match;
```

```

int i,j;
while(1){
    scanf("%d %d", &a[0], &e1);           // a0 と桁数を読み込む
    if(a[0]==0 && e1==0) break;          // 共に 0 なら終了
    match = 0;
    for(i=1; i<=MAX_I; i++){             // i は MAX_I = 20 を超えないと仮定してよい
        a[i] = next_val(a[i-1], e1);     // 次の数列の値 ai を求める
        for(j=0; j<i; j++){              // a0 から a_{i-1} で ai と等しいものを探す
            if(a[j] == a[i]){            // 見つかった
                match = 1;
                break;
            }
        }
        if(match) break;                 // 見つかったので、次の数列の値を求める必要はない
    }
    printf("%d %d %d¥n", j, a[i], i-j); // 答えの出力
}
}

```

- 数列の次の値を求める別の方法
    1. 配列  $dc[i] = 0$  ( $0 \leq i \leq 9$ ) とする。<sup>4</sup>
    2. 10進数の各桁  $d$  に対して、 $dc[d]++$  を実行する。
    3. 配列  $dc$  の各要素の値から、最大値  $max$  と最小値  $min$  を求める。<sup>5</sup>
    4.  $max - min$  が求める値である。
- ※ (バブル)ソートの代わりにバケットソートを行ったとの見方も出来る。



大きい数値から順(9→0)に生起回数個並べると 886320 = 最大値  
 小さい数値から順(0→9)に生起回数個並べると 023688 = 最小値

### 【プログラム例2】

```
// ICPC 2012 国内予選問題 B
//
// Filename = pb2.c
// Compile : cc pb2.c
// Execution : ./a.out < B0 > B0.result
// Check : diff B0.ans B0.result
//
#include <stdio.h>

#define MAX_DIGITS 6
#define MAX_I 20

// 数列の x の次の値を計算して返す。el は桁数。
```

<sup>4</sup> 配列  $dc$  は、10進数表記で0~9の生起回数を数えるための配列である。  
<sup>5</sup> まず、 $max = min = 0$  とする。最大値の場合は  $i = 9 \rightarrow 0$ 、最小値の場合は  $i = 0 \rightarrow 9$  の順に、 $max$  や  $min$  を10倍して  $i$  を加算する操作を  $dc[i]$  回繰り返せば良い。(  $dc[i] = 0$  の時は、10倍して  $i$  を加算する操作は行われぬ。)

```

int next_val(int x, int el)
{
    int dc[10];           // 10 進数の各桁における数値の生起回数を数える配列
    int i,j, tmp;
    int max, min;
    for(i=0; i<10; i++) dc[i]= 0;           // 生起回数を数える配列を 0 に初期化
    for(i=0; i<el; i++){                   // 各桁に 0~9 が何回現れるかを数える
        dc[x%10]++;
        x = x/10;
    }
    // 最大値と最小値を求める
    max = min = 0;
    for(i=0; i<10; i++){
        for(j=1; j <= dc[i]; j++){         // 数値の小さい方が上位桁
            min = min*10 + i;
        }
        for(j=1; j <= dc[9-i]; j++){       // 数値の大きい方が上位桁
            max = max*10 + (9-i);
        }
    }
    return max-min;                       // 数列の次の値
}

int main()
{
    int a[MAX_I + 1];                     // 数列の値を格納する配列
    int el;                               // 桁数を格納する変数
    int match;
    int i,j;
    while(1){
        scanf("%d %d", &a[0], &el);       // a0 と桁数を読み込む
        if(a[0]==0 && el==0) break;       // 共に 0 なら終了
        match = 0;
        for(i=1; i<=MAX_I; i++){          // i は MAX_I = 20 を超えないと仮定してよい
            a[i] = next_val(a[i-1], el);   // 次の数列の値 ai を求める
            for(j=0; j<i; j++){            // a0 から a_{i-1} で ai と等しいものを探す
                if(a[j] == a[i]){         // 見つかった
                    match = 1;
                    break;
                }
            }
        }
        if(match) break;                  // 見つかったので、次の数列の値を求める必要はない
    }
    printf("%d %d %d¥n", j, a[i], i-j);    // 答えの出力
}

```