

MONK e o Algoritmo de Ordenação

Monk recentemente conversou com Fredo sobre ordenação. Agora ele quer ver se Fredo entendeu ou não o conceito. Por isso, ele deu a ele o seguinte algoritmo e pediu para que implementasse:

Consideremos o dígito mais à direita de cada número sendo o índice 1, segundo mais à direita o índice dois e fazemos isso até o mais à esquerda. Significado do *inésimo* pedaço: O pedaço consiste nos dígitos da posição $5 * i$ até $1 + 5 * (i - 1)$ no dado número. Se não há dígito nessa posição, então pegamos 0.

Inicialmente, i é 1.

- Construa o *inésimo* pedaço, para todos os inteiros no vetor. Vamos chamar o valor desse pedaço como sendo o peso do respectivo inteiro no array.
- Se o peso de todos os inteiros no vetor é 0, então PARE.
- Ordene o vetor de acordo com os pesos dos inteiros. Se dois inteiros tem o mesmo peso, então o que apareceu antes deve ser posicionado antes mesmo depois da ordenação. O vetor muda para o vetor ordenado.
- Incremente i por 1 e repita desde o *passo* 1

Veja o exemplo simples para um melhor entendimento.

Então, Fredo entendeu o conceito e codificou. Agora, Monk pediu você para escrever o código para o algoritmo para verificar a resposta de Fredo.

Entrada

A *primeira linha* de entrada contém N denotando o número de elementos no vetor que será sorteado. A *próxima linha* contém N inteiros separados por espaço, denotando os elementos do vetor.

Saída

Você precisa printar o novo vetor em cada passo do algoritmo.

Restrições:

- $1 \leq N \leq 10^6$
- $1 \leq A[i] \leq 10^{18}$; $A[]$ é o array de entrada
- Tamanho dos inteiros em A não são iguais.

Exemplos

Entradas	Saídas
3	213456789 123456789 167890
213456789 167890 123456789	167890 123456789 213456789