



# Heidi's Calling Codes

## Heidi et les Indicatifs Téléphoniques

### Handys mit Chaotischen Codes

Cheng Zhong

#### Hard Task — Prefix

Given a *valid* set of CCCs which is *not full*. Decide the minimum number of CCCs to be added into the set to make it full.

To do so, write a function `int minimalNumberToMakeFull(int n, char* prefix[])` that

- Takes as parameters
  - *n* : the number of CCCs
  - *prefix* : an array with *n* CCCs at positions 0 to *n* − 1
- Returns the minimal number of missing CCCs to make the set *full*.

**Sample Input and Output** For the set {000, 01, 0011, 10}, you should return 2. Actually, the prospective two CCCs could be “0010” and “11”.

#### Tâche Difficile — Préfixe

Étant donné un ensemble *valide* de ITI qui n'est pas *complet*. Déterminez quel est le nombre minimum d'ITI à ajouter à l'ensemble afin de le rendre *complet*.

Pour cela, écrivez une fonction `int minimalNumberToMakeFull(int n, char* prefix[])` qui

- Prend les paramètres suivants
  - *n* : le nombre de ITIs.
  - *prefix* : un vecteur avec *n* ITIs aux positions 0 à *n* − 1
- Retourne le nombre minimum de ITI à ajouter pour rendre l'ensemble *complet*.

**Exemple d'Entrées et Sorties** Pour l'ensemble {000, 01, 0011, 10}, vous devriez retourner 2. Par exemple, deux ITI qui pourraient compléter l'ensemble sont “0010” et “11”.

#### Schwierige Aufgabe — Präfix

Heidi analysiert nun *gültige*, nicht *volle* Mengen CCCs. Berechnet für eine gegebene (*gültige* und nicht *volle*) Menge die minimale Anzahl Codes, die hinzugefügt werden müssen, damit die Menge *voll* wird.

Schreibt dazu eine Funktion `int minimalNumberToMakeFull(int n, char* prefix[])` welche

- Folgende Parameter nimmt
  - *n* : Anzahl der CCCs
  - *prefix* : Ein Array mit *n* CCCs auf Positionen 0 bis *n* − 1
- Die minimale Anzahl an hinzuzufügenden CCCs zurückgibt, damit die Menge *voll* wird.

**Beispieleingabe und -ausgabe** Für die Menge {000, 01, 0011, 10} ist die minimale Anzahl 2. Zwei mögliche CCCs, welche die Menge *voll* machen, sind „0010“ und „11“.