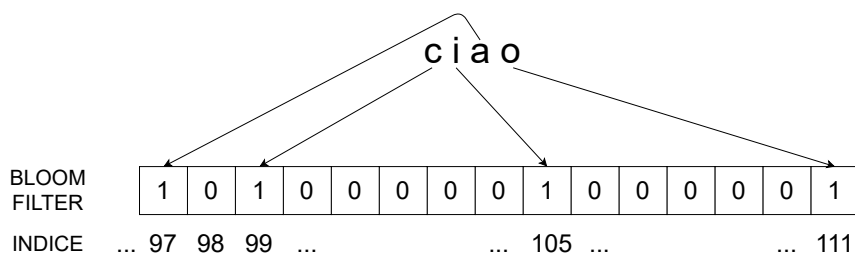


## 1 Esercizio 3<sub>1</sub>

Un "Bloom Filter" è un Tipo di Dato Astratto (TDA) usato in programmazione per controllare velocemente se un certo elemento (per esempio, una parola) appartiene a un determinato insieme di elementi (per esempio, una frase) precedentemente inseriti nel Bloom Filter. Concretamente, un Bloom Filter consiste in un array di dimensione **n** contenente valori booleani. Inizialmente, ogni cella dell'array ha valore 0 ("**false**"). L'inserimento di un elemento consiste nel settare a 1 ("**true**") il valore di alcune celle del Bloom Filter, i cui indici sono derivati da una funzione applicata all'elemento che si vuole inserire. Per esempio, se l'elemento da inserire è una parola, gli indici delle celle del Bloom Filter da settare a 1 potrebbero corrispondere al valore ASCII di ogni singola lettera della parola (vedi l'immagine sotto per un esempio).



Il controllo della presenza di un elemento nel Bloom Filter consiste nel controllare che tutte le celle corrispondenti all'elemento abbiano valore "1". L'operazione di controllo, pur essendo molto veloce, può ritornare falsi positivi, ma mai falsi negativi. In altre parole, l'operazione di controllo, dato un elemento, può ritornare due valori: "l'elemento può forse essere presente nell'insieme" oppure "l'elemento non è sicuramente presente nell'insieme". Questo è dato dal fatto che due elementi diversi potrebbero corrispondere alle stesse celle del Bloom Filter (nell'esempio di sopra, le parole "ciao" e "caio" corrispondono alle stesse celle).

Implementare la struttura dati e le operazioni di un Bloom Filter scrivendo nel file `esercizio3.cc` le dichiarazioni e le definizioni delle seguenti quattro funzioni:

- **init**: questa funzione prende come unico argomento un intero positivo **n**. La funzione **init** deve creare dinamicamente un array di booleani di dimensione **n** (il Bloom Filter), settare tutte le celle dell'array a **false** e ritornare il puntatore all'array appena creato;
- **insert**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri. Per ogni carattere (tranne il carattere terminatore), la funzione **insert** deve settare a "**true**" la cella del Bloom Filter con indice uguale al valore ASCII del carattere. Se il valore ASCII del carattere supera

la dimensione del Bloom Filter, calcolare (iterativamente o ricorsivamente) il modulo del valore ASCII per la dimensione del Bloom Filter e usarlo come indice. Per esempio, se il carattere da inserire è **z** (valore ASCII 122) e la dimensione del Bloom Filter è 100, l'indice della cella da settare a **true** è  $122 \% 100 = 22$ . Se una cella del Bloom Filter ha già valore **true**, ignorarla e procedere con il prossimo carattere dell'array fornito come argomento, se presente;

- **check**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri, e ritorna un valore booleano. la funzione **check** deve controllare se l'array di caratteri è stato precedentemente inserito nel Bloom Filter, ritornare "true" in caso affermativo e "false" altrimenti. Suggerimento: per controllare se un array di caratteri è stato precedentemente inserito nel Bloom Filter, controllare che le corrispondenti celle abbiano valore **true**;
- **deinit**: questa funzione prende come unico argomento un array di booleani (il Bloom Filter). La funzione **deinit** deve deallocare il Bloom Filter.

Questo è un esempio di esecuzione:

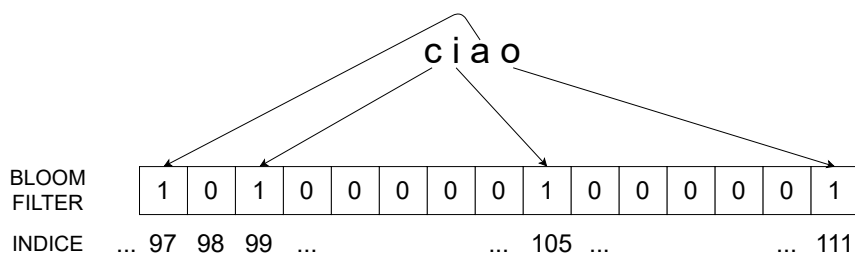
```
computer > ./a.out
Ho inserito la parola 'ciao' nel bloom filter
La parola 'ciao' è presente nel bloom filter
La parola 'volo' NON è presente nel bloom filter
```

**Note:**

- Scaricare il file **esercizio3.cc**, modificarlo per inserire la corretta definizione e implementazione delle funzioni **init**, **insert**, **check** e **deinit** e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo **static** e di funzioni di libreria al di fuori di quelle definite in **iostream** e **cstdlib**.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

## 2 Esercizio 3<sub>2</sub>

Un "Bloom Filter" è un Tipo di Dato Astratto (TDA) usato in programmazione per controllare velocemente se un certo elemento (per esempio, una parola) appartiene a un determinato insieme di elementi (per esempio, una frase) precedentemente inseriti nel Bloom Filter. Concretamente, un Bloom Filter consiste in un array di dimensione **n** contenente valori booleani. Inizialmente, ogni cella dell'array ha valore 0 ("**false**"). L'inserimento di un elemento consiste nel settare a 1 ("**true**") il valore di alcune celle del Bloom Filter, i cui indici sono derivati da una funzione applicata all'elemento che si vuole inserire. Per esempio, se l'elemento da inserire è una parola, gli indici delle celle del Bloom Filter da settare a 1 potrebbero corrispondere al valore ASCII di ogni singola lettera della parola (vedi l'immagine sotto per un esempio).



Il controllo della presenza di un elemento nel Bloom Filter consiste nel controllare che tutte le celle corrispondenti all'elemento abbiano valore "1". L'operazione di controllo, pur essendo molto veloce, può ritornare falsi positivi, ma mai falsi negativi. In altre parole, l'operazione di controllo, dato un elemento, può ritornare due valori: "l'elemento può forse essere presente nell'insieme" oppure "l'elemento non è sicuramente presente nell'insieme". Questo è dato dal fatto che due elementi diversi potrebbero corrispondere alle stesse celle del Bloom Filter (nell'esempio di sopra, le parole "ciao" e "caio" corrispondono alle stesse celle).

Implementare la struttura dati e le operazioni di un Bloom Filter scrivendo nel file `esercizio3.cc` le dichiarazioni e le definizioni delle seguenti quattro funzioni:

- **create**: questa funzione prende come unico argomento un intero positivo **n**. La funzione **create** deve creare dinamicamente un array di booleani di dimensione **n** (il Bloom Filter), settare tutte le celle dell'array a **false** e ritornare il puntatore all'array appena creato;
- **add**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri. Per ogni carattere (tranne il carattere terminatore), la funzione **add** deve settare a "**true**" la cella del Bloom Filter con indice uguale al valore ASCII del carattere. Se il valore ASCII del carattere supera

la dimensione del Bloom Filter, calcolare (iterativamente o ricorsivamente) il modulo del valore ASCII per la dimensione del Bloom Filter e usarlo come indice. Per esempio, se il carattere da inserire è **z** (valore ASCII 122) e la dimensione del Bloom Filter è 100, l'indice della cella da settare a **true** è  $122 \% 100 = 22$ . Se una cella del Bloom Filter ha già valore **true**, ignorarla e procedere con il prossimo carattere dell'array fornito come argomento, se presente;

- **control**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri, e ritorna un valore booleano. la funzione **control** deve controllare se l'array di caratteri è stato precedentemente inserito nel Bloom Filter, ritornare "true" in caso affermativo e "false" altrimenti. Suggerimento: per controllare se un array di caratteri è stato precedentemente inserito nel Bloom Filter, controllare che le corrispondenti celle abbiano valore **true**;
- **destroy**: questa funzione prende come unico argomento un array di booleani (il Bloom Filter). La funzione **destroy** deve deallocare il Bloom Filter.

Questo è un esempio di esecuzione:

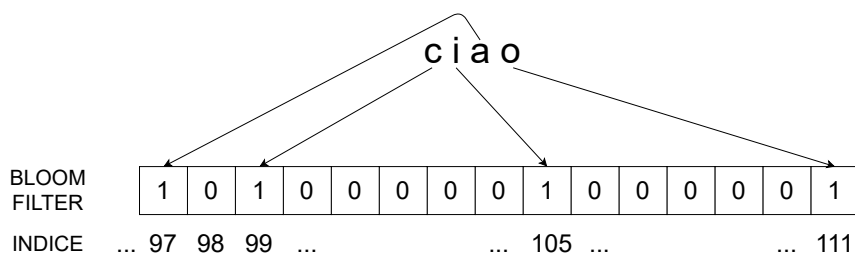
```
computer > ./a.out
Ho inserito la parola 'ciao' nel bloom filter
La parola 'ciao' è presente nel bloom filter
La parola 'volo' NON è presente nel bloom filter
```

**Note:**

- Scaricare il file **esercizio3.cc**, modificarlo per inserire la corretta definizione e implementazione delle funzioni **create**, **add**, **control** e **destroy** e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo **static** e di funzioni di libreria al di fuori di quelle definite in **iostream** e **cstdlib**.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

### 3 Esercizio 3<sub>3</sub>

Un "Bloom Filter" è un Tipo di Dato Astratto (TDA) usato in programmazione per controllare velocemente se un certo elemento (per esempio, una parola) appartiene a un determinato insieme di elementi (per esempio, una frase) precedentemente inseriti nel Bloom Filter. Concretamente, un Bloom Filter consiste in un array di dimensione **n** contenente valori booleani. Inizialmente, ogni cella dell'array ha valore 0 ("**false**"). L'inserimento di un elemento consiste nel settare a 1 ("**true**") il valore di alcune celle del Bloom Filter, i cui indici sono derivati da una funzione applicata all'elemento che si vuole inserire. Per esempio, se l'elemento da inserire è una parola, gli indici delle celle del Bloom Filter da settare a 1 potrebbero corrispondere al valore ASCII di ogni singola lettera della parola (vedi l'immagine sotto per un esempio).



Il controllo della presenza di un elemento nel Bloom Filter consiste nel controllare che tutte le celle corrispondenti all'elemento abbiano valore "1". L'operazione di controllo, pur essendo molto veloce, può ritornare falsi positivi, ma mai falsi negativi. In altre parole, l'operazione di controllo, dato un elemento, può ritornare due valori: "l'elemento può forse essere presente nell'insieme" oppure "l'elemento non è sicuramente presente nell'insieme". Questo è dato dal fatto che due elementi diversi potrebbero corrispondere alle stesse celle del Bloom Filter (nell'esempio di sopra, le parole "ciao" e "caio" corrispondono alle stesse celle).

Implementare la struttura dati e le operazioni di un Bloom Filter scrivendo nel file `esercizio3.cc` le dichiarazioni e le definizioni delle seguenti quattro funzioni:

- **init**: questa funzione prende come unico argomento un intero positivo **n**. La funzione **init** deve creare dinamicamente un array di booleani di dimensione **n** (il Bloom Filter), settare tutte le celle dell'array a **false** e ritornare il puntatore all'array appena creato;
- **add**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri. Per ogni carattere (tranne il carattere terminatore), la funzione **add** deve settare a "**true**" la cella del Bloom Filter con indice uguale al valore ASCII del carattere. Se il valore ASCII del carattere supera

la dimensione del Bloom Filter, calcolare (iterativamente o ricorsivamente) il modulo del valore ASCII per la dimensione del Bloom Filter e usarlo come indice. Per esempio, se il carattere da inserire è **z** (valore ASCII 122) e la dimensione del Bloom Filter è 100, l'indice della cella da settare a **true** è  $122 \% 100 = 22$ . Se una cella del Bloom Filter ha già valore **true**, ignorarla e procedere con il prossimo carattere dell'array fornito come argomento, se presente;

- **check**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri, e ritorna un valore booleano. la funzione **check** deve controllare se l'array di caratteri è stato precedentemente inserito nel Bloom Filter, ritornare "true" in caso affermativo e "false" altrimenti. Suggerimento: per controllare se un array di caratteri è stato precedentemente inserito nel Bloom Filter, controllare che le corrispondenti celle abbiano valore **true**;
- **destroy**: questa funzione prende come unico argomento un array di booleani (il Bloom Filter). La funzione **destroy** deve deallocare il Bloom Filter.

Questo è un esempio di esecuzione:

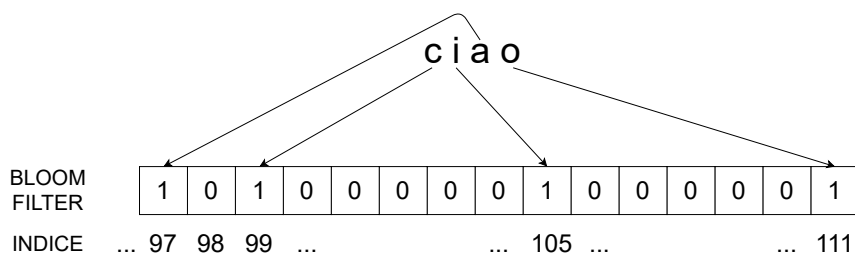
```
computer > ./a.out
Ho inserito la parola 'ciao' nel bloom filter
La parola 'ciao' è presente nel bloom filter
La parola 'volo' NON è presente nel bloom filter
```

**Note:**

- Scaricare il file **esercizio3.cc**, modificarlo per inserire la corretta definizione e implementazione delle funzioni **init**, **add**, **check** e **destroy** e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo **static** e di funzioni di libreria al di fuori di quelle definite in **iostream** e **cstdlib**.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.

## 4 Esercizio 3<sub>4</sub>

Un "Bloom Filter" è un Tipo di Dato Astratto (TDA) usato in programmazione per controllare velocemente se un certo elemento (per esempio, una parola) appartiene a un determinato insieme di elementi (per esempio, una frase) precedentemente inseriti nel Bloom Filter. Concretamente, un Bloom Filter consiste in un array di dimensione **n** contenente valori booleani. Inizialmente, ogni cella dell'array ha valore 0 ("**false**"). L'inserimento di un elemento consiste nel settare a 1 ("**true**") il valore di alcune celle del Bloom Filter, i cui indici sono derivati da una funzione applicata all'elemento che si vuole inserire. Per esempio, se l'elemento da inserire è una parola, gli indici delle celle del Bloom Filter da settare a 1 potrebbero corrispondere al valore ASCII di ogni singola lettera della parola (vedi l'immagine sotto per un esempio).



Il controllo della presenza di un elemento nel Bloom Filter consiste nel controllare che tutte le celle corrispondenti all'elemento abbiano valore "1". L'operazione di controllo, pur essendo molto veloce, può ritornare falsi positivi, ma mai falsi negativi. In altre parole, l'operazione di controllo, dato un elemento, può ritornare due valori: "l'elemento può forse essere presente nell'insieme" oppure "l'elemento non è sicuramente presente nell'insieme". Questo è dato dal fatto che due elementi diversi potrebbero corrispondere alle stesse celle del Bloom Filter (nell'esempio di sopra, le parole "ciao" e "caio" corrispondono alle stesse celle).

Implementare la struttura dati e le operazioni di un Bloom Filter scrivendo nel file `esercizio3.cc` le dichiarazioni e le definizioni delle seguenti quattro funzioni:

- **create**: questa funzione prende come unico argomento un intero positivo **n**. La funzione **create** deve creare dinamicamente un array di booleani di dimensione **n** (il Bloom Filter), settare tutte le celle dell'array a **false** e ritornare il puntatore all'array appena creato;
- **insert**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri. Per ogni carattere (tranne il carattere terminatore), la funzione **insert** deve settare a "**true**" la cella del Bloom Filter con indice uguale al valore ASCII del carattere. Se il valore ASCII del carattere supera

la dimensione del Bloom Filter, calcolare (iterativamente o ricorsivamente) il modulo del valore ASCII per la dimensione del Bloom Filter e usarlo come indice. Per esempio, se il carattere da inserire è **z** (valore ASCII 122) e la dimensione del Bloom Filter è 100, l'indice della cella da settare a **true** è  $122 \% 100 = 22$ . Se una cella del Bloom Filter ha già valore **true**, ignorarla e procedere con il prossimo carattere dell'array fornito come argomento, se presente;

- **control**: questa funzione prende come argomenti un array di booleani (il Bloom Filter), la dimensione del Bloom Filter **n** e un array di caratteri, e ritorna un valore booleano. la funzione **control** deve controllare se l'array di caratteri è stato precedentemente inserito nel Bloom Filter, ritornare "true" in caso affermativo e "false" altrimenti. Suggerimento: per controllare se un array di caratteri è stato precedentemente inserito nel Bloom Filter, controllare che le corrispondenti celle abbiano valore **true**;
- **deinit**: questa funzione prende come unico argomento un array di booleani (il Bloom Filter). La funzione **deinit** deve deallocare il Bloom Filter.

Questo è un esempio di esecuzione:

```
computer > ./a.out
Ho inserito la parola 'ciao' nel bloom filter
La parola 'ciao' è presente nel bloom filter
La parola 'volo' NON è presente nel bloom filter
```

**Note:**

- Scaricare il file **esercizio3.cc**, modificarlo per inserire la corretta definizione e implementazione delle funzioni **create**, **insert**, **control** e **deinit** e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- E' consentito definire e implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo **static** e di funzioni di libreria al di fuori di quelle definite in **iostream** e **cstdlib**.
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione.