

## 1 Esercizio 1

La conversione di un numero in base  $B$  in decimale è la trasformazione di un numero in base  $B$  (sistema in base  $B$ , e.g. 2) in un numero in base 10 (sistema decimale). Essendo due sistemi numerici posizionali per convertire un numero in base  $B$  in un numero decimale è sufficiente moltiplicare ogni cifra  $d_i$  del numero  $d_n d_{n-1} \dots d_1 d_0$  in base  $B$  per la potenza  $B$  corrispondente alla sua posizione ( $n$ ), ovvero il numero decimale corrispondente è:  $d_n * B^n + d_{n-1} * B^{n-1} + \dots + d_1 * B^1 + d_0 * B^0$ .

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della **funzione ricorsiva** `converti` che prende come argomento un array di `char source`, la dimensione massima dell'array `source` di tipo `int`, e assumendo  $B = 2$  e che gli elementi dell'array siano i soli caratteri `'0'`, `'1'` o `'\0'` per indicare la fine dell'array, ritorni il numero intero (di tipo `long long int`) che rappresenta la conversione in decimale del numero in base  $B$  rappresentato nell'array di caratteri `source` sfruttando il metodo sopra descritto.

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output il numero decimale convertito dalla base specificata.

Questo è un esempio di esecuzione:

```
computer > ./a.out
Inserire numero: 111
The converted number is: 7

computer > echo 1101 | ./a.out
Inserire numero: The converted number is: 13

computer > echo 10010110000 | ./a.out
Inserire numero: The converted number is: 1200
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `converti`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `converti` deve essere ricorsiva ed al suo interno **NON ci possono essere cicli o chiamate a funzioni contenenti cicli**. Si può però fare uso di funzioni ausiliarie da chiamare all'interno di questa funzione che NON contengano cicli o che siano ricorsive.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- 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 `cstddef`. In particolare **non sono ammesse** funzioni definite in `cctype` (e.g. `tolower`, `isdigit`, ...).
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si suggerisce di scorrere ricorsivamente l'array in modo che la ricorsione calcoli la somma corrispondente ai caratteri rimanenti: il caso base consiste nell'incontrare il carattere `'\0'`

che ritorna 0, per il carattere  $i$ -esimo si ritorna il valore risultante dalla chiamata ricorsiva addizionato al valore corrispondente alla lettera sotto analisi (i.e.,  $d_i * B^i$ ).

## 2 Esercizio 2

La conversione di un numero in base  $B$  in decimale è la trasformazione di un numero in base  $B$  (sistema in base  $B$ , e.g. 2) in un numero in base 10 (sistema decimale). Essendo due sistemi numerici posizionali per convertire un numero in base  $B$  in un numero decimale è sufficiente moltiplicare ogni cifra  $d_i$  del numero  $d_n d_{n-1} \dots d_1 d_0$  in base  $B$  per la potenza  $B$  corrispondente alla sua posizione ( $n$ ), ovvero il numero decimale corrispondente è:  $d_n * B^n + d_{n-1} * B^{n-1} + \dots + d_1 * B^1 + d_0 * B^0$ .

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della **funzione ricorsiva** `converti` che prende come argomento un array di `char source`, la dimensione massima dell'array `source` di tipo `int`, e assumendo  $B = 4$  e che gli elementi dell'array siano i soli caratteri `'0'`, `'1'`, `'2'`, `'3'` o `'\0'` per indicare la fine dell'array, ritorni il numero intero (di tipo `long long int`) che rappresenta la conversione in decimale del numero in base  $B$  rappresentato nell'array di caratteri `source` sfruttando il metodo sopra descritto.

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output il numero decimale convertito dalla base specificata.

Questo è un esempio di esecuzione:

```
computer > ./a.out
Inserire numero: 33
The converted number is: 15

computer > echo 300 | ./a.out
Inserire numero: The converted number is: 48

computer > echo 10010110000 | ./a.out
Inserire numero: The converted number is: 1066240
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `converti`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `converti` deve essere ricorsiva ed al suo interno **NON ci possono essere cicli o chiamate a funzioni contenenti cicli**. Si può però fare uso di funzioni ausiliarie da chiamare all'interno di questa funzione che NON contengano cicli o che siano ricorsive.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- 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 `cstddef`. In particolare **non sono ammesse** funzioni definite in `cctype` (e.g. `tolower`, `isdigit`), ...).

- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si suggerisce di scorrere ricorsivamente l'array in modo che la ricorsione calcoli la somma corrispondente ai caratteri rimanenti: il caso base consiste nell'incontrare il carattere '\0' che ritorna 0, per il carattere  $i$ -esimo si ritorna il valore risultante dalla chiamata ricorsiva addizionato al valore corrispondente alla lettera sotto analisi (i.e.,  $d_i * B^i$ ).

### 3 Esercizio 3

La conversione di un numero in base  $B$  in decimale è la trasformazione di un numero in base  $B$  (sistema in base  $B$ , e.g. 2) in un numero in base 10 (sistema decimale). Essendo due sistemi numerici posizionali per convertire un numero in base  $B$  in un numero decimale è sufficiente moltiplicare ogni cifra  $d_i$  del numero  $d_n d_{n-1} \dots d_1 d_0$  in base  $B$  per la potenza  $B$  corrispondente alla sua posizione ( $n$ ), ovvero il numero decimale corrispondente è:  $d_n * B^n + d_{n-1} * B^{n-1} + \dots + d_1 * B^1 + d_0 * B^0$ .

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della **funzione ricorsiva** `converti` che prende come argomento un array di `char source`, la dimensione massima dell'array `source` di tipo `int`, e assumendo  $B = 8$  e che gli elementi dell'array siano i soli caratteri '0', '1', '2', '3', '4', '5', '6', '7' o '\0' per indicare la fine dell'array, ritorni il numero intero (di tipo `long long int`) che rappresenta la conversione in decimale del numero in base  $B$  rappresentato nell'array di caratteri `source` sfruttando il metodo sopra descritto.

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output il numero decimale convertito dalla base specificata.

Questo è un esempio di esecuzione:

```
computer > ./a.out
Inserire numero: 33
The converted number is: 27

computer > echo 300 | ./a.out
Inserire numero: The converted number is: 192

computer > echo 10010110000 | ./a.out
Inserire numero: The converted number is: 1075875840
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `converti`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `converti` deve essere ricorsiva ed al suo interno **NON ci possono essere cicli o chiamate a funzioni contenenti cicli**. Si può però fare uso di funzioni ausiliarie da chiamare all'interno di questa funzione che NON contengano cicli o che siano ricorsive.
- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).

- 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 `cstddef`. In particolare **non sono ammesse** funzioni definite in `cctype` (e.g. `tolower`, `isdigit`), ...).
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si suggerisce di scorrere ricorsivamente l'array in modo che la ricorsione calcoli la somma corrispondente ai caratteri rimanenti: il caso base consiste nell'incontrare il carattere `'\0'` che ritorna 0, per il carattere  $i$ -esimo si ritorna il valore risultante dalla chiamata ricorsiva addizionato al valore corrispondente alla lettera sotto analisi (i.e.,  $d_i * B^i$ ).

## 4 Esercizio 3

La conversione di un numero in base  $B$  in decimale è la trasformazione di un numero in base  $B$  (sistema in base  $B$ , e.g. 2) in un numero in base 10 (sistema decimale). Essendo due sistemi numerici posizionali per convertire un numero in base  $B$  in un numero decimale è sufficiente moltiplicare ogni cifra  $d_i$  del numero  $d_n d_{n-1} \dots d_1 d_0$  in base  $B$  per la potenza  $B$  corrispondente alla sua posizione ( $n$ ), ovvero il numero decimale corrispondente è:  $d_n * B^n + d_{n-1} * B^{n-1} + \dots + d_1 * B^1 + d_0 * B^0$ .

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della **funzione ricorsiva** `converti` che prende come argomento un array di `char source`, la dimensione massima dell'array `source` di tipo `int`, e assumendo  $B = 16$  e che gli elementi dell'array siano i soli caratteri `'0'`, `'1'`, `'2'`, `'3'`, `'4'`, `'5'`, `'6'`, `'7'`, `'8'`, `'9'`, `'a'`, `'b'`, `'c'`, `'d'`, `'e'`, ed `'f'` o `'\0'` per indicare la fine dell'array, ritorni il numero intero (di tipo `long long int`) che rappresenta la conversione in decimale del numero in base  $B$  rappresentato nell'array di caratteri `source` sfruttando il metodo sopra descritto. (Assumere che le lettere `'a'`, `'b'`, `'c'`, `'d'` siano sempre minuscole, e che `'a'` = 10, `'b'` = 11, `'c'` = 12, `'d'` = 13 e `'e'` = 14, `'f'` = 15.)

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output il numero decimale convertito dalla base specificata.

Questo è un esempio di esecuzione:

```
computer > ./a.out
Inserire numero: 33
The converted number is: 51

computer > echo 33a | ./a.out
Inserire numero: The converted number is: 826

computer > echo 1aa2d000 | ./a.out
Inserire numero: The converted number is: 446877696
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `converti`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `converti` deve essere ricorsiva ed al suo interno **NON ci possono essere cicli o chiamate a funzioni contenenti cicli**. Si può però fare uso di funzioni ausiliarie da chiamare all'interno di questa funzione che NON contengano cicli o che siano ricorsive.

- Le uniche assunzioni che si possono fare sull'input e su dimensioni di eventuali strutture/array utilizzate nel file di partenza fornito sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- 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 **cstddef**. In particolare **non sono ammesse** funzioni definite in **cctype** (e.g. `tolower`, `isdigit`, ...).
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si suggerisce di scorrere ricorsivamente l'array in modo che la ricorsione calcoli la somma corrispondente ai caratteri rimanenti: il caso base consiste nell'incontrare il carattere `'\0'` che ritorna 0, per il carattere  $i$ -esimo si ritorna il valore risultante dalla chiamata ricorsiva addizionato al valore corrispondente alla lettera sotto analisi (i.e.,  $d_i * B^i$ ).