

## 1 Esercizio 1

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della funzione ricorsiva `get_elements` che prende come argomento un array di `char` `source`, la dimensione dell'array `source`, un array `dest1` di `char`, la dimensione massima dell'array `dest1`, la dimensione da calcolare dell'array `dest1`, un array `dest2` di `char`, la dimensione massima dell'array `dest2`, la dimensione da calcolare dell'array `dest2`. Tale funzione:

- estrae dall'array `source` le lettere minuscole e le memorizza nell'ordine in cui compaiono nell'array `dest1` dopo averle convertite nella corrispondente lettera maiuscola;
- estrae dall'array `source` i caratteri numerici e li memorizza nell'ordine in cui compaiono nell'array `dest2` in modo che '9' sia convertito in '0', '8' in '1', ... , '1' in '8', e '0' in '9';
- ignora le lettere maiuscole ed eventuali altri caratteri;
- calcola le dimensioni correnti degli array `dest1` e `dest2`.

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output la sequenza di caratteri letta, e il contenuto degli array risultato della chiamata a `get_elements`.

Questo è un esempio di esecuzione:

```
computer > ./a.out
a b 0 1 A a j K 3 9
Source = a b 0 1 A a j K 3 9
D1      = A B A J
D2      = 9 8 6 0
```

**Note:**

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

## 2 Esercizio 2

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della funzione ricorsiva `get_elements` che prende come argomento un array di `char` `source`, la dimensione dell'array `source`, un array `dest1` di `char`, la dimensione massima dell'array `dest1`, la dimensione da calcolare dell'array `dest1`, un array `dest2` di `char`, la dimensione massima dell'array `dest2`, la dimensione da calcolare dell'array `dest2`. Tale funzione:

- estrae dall'array `source` le lettere minuscole e le memorizza nell'ordine in cui compaiono nell'array `dest1` dopo averle convertite nella corrispondente lettera maiuscola;
- estrae dall'array `source` le lettere maiuscole e le memorizza nell'ordine in cui compaiono nell'array `dest2` dopo averle convertite nella corrispondente lettera minuscola;
- ignora ogni altro tipo di carattere (e.g. numeri o punteggiature);
- calcola le dimensioni correnti degli array `dest1` e `dest2`.

Il programma per essere eseguito legge da standard input una sequenza di caratteri terminata da newline e produce in output la sequenza di caratteri letta, e il contenuto degli array risultato della chiamata a `get_elements`.

Questo è un esempio di esecuzione:

```
computer > ./a.out
a b c D s A a y U 6 8
Source = a b c D s A a y U 6
D1      = A B C S A Y
D2      = d a u
```

**Note:**

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

### 3 Esercizio 3

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della funzione ricorsiva `get_elements` che prende come argomento un array di `int` `source`, la dimensione dell'array `source`, un array `dest1` di `int`, la dimensione massima dell'array `dest1`, la dimensione da calcolare dell'array `dest1`, un array `dest2` di `int`, la dimensione massima dell'array `dest2`, la dimensione da calcolare dell'array `dest2`. Tale funzione:

- estrae dall'array `source` gli elementi che si trovano in posizione pari e che sono multipli di 3 e li memorizza nell'ordine in cui compaiono nell'array `dest1`;
- estrae dall'array `source` gli altri elementi (che non sono in posizione pari o multipli di 3) e li memorizza nell'ordine in cui compaiono nell'array `dest2`;
- calcola le dimensioni correnti degli array `dest1` e `dest2`.

Il programma per essere eseguito legge da standard input una sequenza di numeri terminata da -1 e produce in output la sequenza di caratteri letta, e il contenuto degli array risultato della chiamata a `get_elements`.

Questo è un esempio di esecuzione:

```
computer > ./a.out
1 2 3 5 8 9 22 24 9 8 -1
Source = 1 2 3 5 8 9 22 24 9 8
D1      = 3 9
D2      = 1 2 5 8 9 22 24 8
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `get_elements`, e caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- Si consideri lo 0 come pari.
- La funzione `get_elements` deve essere ricorsiva ed al suo interno non ci possono essere cicli o chiamate a funzioni contenenti cicli. Si può però fare uso di eventuali funzioni ricorsive ausiliarie all'interno di questa funzione.
- 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 `ctypes`.

## 4 Esercizio 4

Scrivere nel file `esercizio2.cc` la dichiarazione e la definizione della funzione ricorsiva `get_elements` che prende come argomento un array di `int` `source`, la dimensione dell'array `source`, un array `dest1` di `int`, la dimensione massima dell'array `dest1`, la dimensione da calcolare dell'array `dest1`, un array `dest2` di `int`, la dimensione massima dell'array `dest2`, la dimensione da calcolare dell'array `dest2`. Tale funzione:

- estrae dall'array `source` gli elementi che si trovano in posizione multiple di 3 e che hanno un valore pari e li memorizza nell'ordine in cui compaiono nell'array `dest1` dopo averlo moltiplicato per 2;
- estrae dall'array `source` gli altri elementi (che non sono in posizione multiple di 3 o che non hanno un valore pari) e li memorizza nell'ordine in cui compaiono nell'array `dest2` dopo averli moltiplicati per 3;
- calcola le dimensioni correnti degli array `dest1` e `dest2`.

Il programma per essere eseguito legge da standard input una sequenza di numeri terminata da -1 e produce in output la sequenza di caratteri letta, e il contenuto degli array risultato della chiamata a `get_elements`.

Questo è un esempio di esecuzione:

```
computer > ./a.out
1 2 3 5 8 9 22 24 9 8 -1
Source = 1 2 3 5 8 9 22 24 9 8
D1      = 44 16
D2      = 3 6 9 15 24 27 72 27
```

**Note:**

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `get_elements`, e caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- Si consideri lo 0 come pari.
- La funzione `get_elements` deve essere ricorsiva ed al suo interno non ci possono essere cicli o chiamate a funzioni contenenti cicli. Si può però fare uso di eventuali funzioni ricorsive ausiliarie all'interno di questa funzione.
- 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 `ctypes`.