

1 Esercizio 1

Scrivere la dichiarazione e la definizione di una funzione **ricorsiva** `f`, che prende come argomento un intero `n` e ritorna il numero di coppie di cifre uguali in posizioni adiacenti presenti nel numero `n`, nel caso `n` sia negativo restituisce 0. La funzione è inserita in un semplice programma che prende l'intero da analizzare come primo argomento del comando, come illustrato nei seguenti esempi di esecuzione:

```
computer > ./a.out 551122
The read string is: 551122
The converted value is: 551122
function(551122) = 3

computer > ./a.out 55511222
The read string is: 55511222
The converted value is: 55511222
function(55511222) = 5

computer > ./a.out 1232123
The read string is: 1232123
The converted value is: 1232123
function(1232123) = 0
```

Il `main` è già fornito, e non necessita di essere modificato.

Note:

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `funzione`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `funzione` 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 `cstdint`.
- 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.

2 Esercizio 2

Scrivere la dichiarazione e la definizione di una funzione **ricorsiva** `f`, che prende come argomento un intero `n` e ritorna il numero di coppie di cifre differenti in posizioni adiacenti presenti nel numero `n`, nel caso `n` sia negativo restituisce 0. La funzione è inserita in un semplice programma che prende l'intero da analizzare come primo argomento del comando, come illustrato nei seguenti esempi di esecuzione:

```

computer > ./a.out 551122
The read string is: 551122
The converted value is: 551122
function(551122) = 2

computer > ./a.out 55511222
The read string is: 55511222
The converted value is: 55511222
function(55511222) = 2

computer > ./a.out 1232123
The read string is: 1232123
The converted value is: 1232123
function(1232123) = 6

```

Il main è già fornito, e non necessita di essere modificato.

Note:

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `funzione`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `funzione` 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 `cstdint`.
- 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.

3 Esercizio 3

Scrivere la dichiarazione e la definizione di una funzione **ricorsiva** `f`, che prende come argomento un intero `n` e ritorna il numero di coppie di cifre pari in posizioni adiacenti presenti nel numero `n`, nel caso `n` sia negativo restituisce 0. La funzione è inserita in un semplice programma che prende l'intero da analizzare come primo argomento del comando, come illustrato nei seguenti esempi di esecuzione:

```

computer > ./a.out 551122
The read string is: 551122
The converted value is: 551122
function(551122) = 1

```

```
computer > ./a.out 55511222
The read string is: 55511222
The converted value is: 55511222
function(55511222) = 2
```

```
computer > ./a.out 1232123
The read string is: 1232123
The converted value is: 1232123
function(1232123) = 0
```

Il main è già fornito, e non necessita di essere modificato.

Note:

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `funzione`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `funzione` 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 `cstdint`.
- 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.

4 Esercizio 4

Scrivere la dichiarazione e la definizione di una funzione **ricorsiva** `f`, che prende come argomento un intero `n` e ritorna il numero di coppie di cifre dispari in posizioni adiacenti presenti nel numero `n`, nel caso `n` sia negativo restituisce 0. La funzione è inserita in un semplice programma che prende l'intero da analizzare come primo argomento del comando, come illustrato nei seguenti esempi di esecuzione:

```
computer > ./a.out 551122
The read string is: 551122
The converted value is: 551122
function(551122) = 3
```

```
computer > ./a.out 55511222
The read string is: 55511222
The converted value is: 55511222
function(55511222) = 4
```

```
computer > ./a.out 1232123
The read string is: 1232123
The converted value is: 1232123
function(1232123) = 0
```

Il main è già fornito, e non necessita di essere modificato.

Note:

- Scaricare il file `esercizio2.cc`, modificarlo per inserire la dichiarazione e la definizione della funzione `funzione`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- La funzione `funzione` 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`.
- 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.