

Esame 20220111-a

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Sia `input.txt` un file contenente per ogni riga una semplice espressione matematica nel formato `<numero> <operazione> <numero>`. I numeri sono interi positivi in base-17. L'operazione può essere invece una delle seguenti: somma (+), moltiplicazione (*) e potenza (^). I numeri sono separati dall'operazione da un singolo spazio.

```
74 + 21
43G * 5F
DD ^ 2
```

Scrivere nel file `eserciziol.cc` un programma che, dati come argomenti da riga di comando il file contenente le espressioni matematiche e un file in output, scriva sul file in output i risultati delle operazioni matematiche in **base-10**, uno per riga. Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Il programma dovrà poter essere chiamato nel seguente modo:

```
./eserciziol.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
158
122300
54756
```

Note:

- Si assuma che le espressioni matematiche non contengano errori (e.g., codifiche errate, operazioni inesistenti, divisioni per zero, etc.). Si assuma anche che le espressioni matematiche siano separate da un carattere `\n`. Il numero massimo di cifre di ogni numero nel file di input è 10.
- Si ricorda che la base-17 è formata da numeri (0-9) e da lettere (A-G). Le lettere corrispondono ai valori che vanno da 10 a 16 ($A = 10, B = 11, \dots, G = 16$). La procedura di conversione è simile al caso binario, con la differenza della base b . Assumendo come esempio $e = 43G$, $decimale = \sum_{i=0} e[i] \cdot b^i$, dove $e[0] = G$, $e[1] = 3$ ed $e[2] = 4$.
- E' consentito l'utilizzo della funzione `int strlen (const char * str)` della libreria `<cstring>`.
- E' consentito l'utilizzo della funzione `double pow (double base, double exponent)` della libreria `<cmath>` per calcolare la potenza.
- Non è consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- 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`.
- 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.

Information for graders:

(2) Esercizio 1 v2

ESSAY

marked out of 10

penalty 0

File picker

Sia `input.txt` un file contenente per ogni riga una semplice espressione matematica nel formato `<numero> <operazione> <numero>`. I numeri sono interi positivi in base-19. L'operazione può essere invece una delle seguenti: somma (+), moltiplicazione (*) e potenza (^). I numeri sono separati dall'operazione da un singolo spazio.

```
69 + 1G
377 * 55
C6 ^ 2
```

Scrivere nel file `eserciziol.cc` un programma che, dati come argomenti da riga di comando il file contenente le espressioni matematiche e un file in output, scriva sul file in output i risultati delle operazioni matematiche in **base-10**, uno per riga. Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Il programma dovrà poter essere chiamato nel seguente modo:

```
./eserciziol.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
158
112300
54756
```

Note:

- Si assuma che le espressioni matematiche non contengano errori (e.g., codifiche errate, operazioni inesistenti, divisioni per zero, etc.). Si assuma anche che le espressioni matematiche siano separate da un carattere `\n`. Il numero massimo di cifre di ogni numero nel file di input è 10.
- Si ricorda che la base-19 è formata da numeri (0-9) e da lettere (A-I). Le lettere corrispondono ai valori che vanno da 10 a 16 ($A = 10, B = 11, \dots, I = 18$). La procedura di conversione è simile al caso binario, con la differenza della base b . Assumendo come esempio $e = 43G$, $decimale = \sum_{i=0} e[i] \cdot b^i$, dove $e[0] = G$, $e[1] = 3$ ed $e[2] = 4$.
- E' consentito l'utilizzo della funzione `int strlen (const char * str)` della libreria `<cstring>`.
- E' consentito l'utilizzo della funzione `double pow (double base, double exponent)` della libreria `<cmath>` per calcolare la potenza.
- Non è consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- 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`.

- 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.

Information for graders:

(3) Esercizio 1 v3

ESSAY

marked out of 10

penalty 0

File picker

Sia `input.txt` un file contenente per ogni riga una semplice espressione matematica nel formato `<numero> <operazione> <numero>`. I numeri sono interi positivi in base-17. L'operazione può essere invece una delle seguenti: sottrazione (`-`), divisione (`/`) e modulo (`%`). I numeri sono separati dall'operazione da un singolo spazio.

```
74 - 21
43G / 5F
DD % 2
```

Scrivere nel file `eserciziol.cc` un programma che, dati come argomenti da riga di comando il file contenente le espressioni matematiche e un file in output, scriva sul file in output i risultati delle operazioni matematiche in **base-10**, uno per riga. Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Il programma dovrà poter essere chiamato nel seguente modo:

```
./eserciziol.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
88
12
0
```

Note:

- Si assuma che le espressioni matematiche non contengano errori (e.g., codifiche errate, operazioni inesistenti, divisioni per zero, etc.). Si assuma anche che le espressioni matematiche siano separate da un carattere `\n`. Il numero massimo di cifre di ogni numero nel file di input è 10.
- Si ricorda che la base-17 è formata da numeri (0-9) e da lettere (A-G). Le lettere corrispondono ai valori che vanno da 10 a 16 ($A = 10, B = 11, \dots, G = 16$). La procedura di conversione è simile al caso binario, con la differenza della base b . Assumendo come esempio $e = 43G$, $decimale = \sum_{i=0} e[i] \cdot b^i$, dove $e[0] = G$, $e[1] = 3$ ed $e[2] = 4$.
- E' consentito l'utilizzo della funzione `int strlen (const char * str)` della libreria `<cstring>`.
- Non è consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- 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`.
- 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.

Information for graders:

(4) Esercizio 1 v4

ESSAY

marked out of 10

penalty 0

File picker

Sia `input.txt` un file contenente per ogni riga una semplice espressione matematica nel formato `<numero> <operazione> <numero>`. I numeri sono interi positivi in base-19. L'operazione può essere invece una delle seguenti: sottrazione (`-`), divisione (`/`) e modulo (`%`). I numeri sono separati dall'operazione da un singolo spazio.

```
69 - 1G
377 / 55
C6 % 2
```

Scrivere nel file `eserciziol.cc` un programma che, dati come argomenti da riga di comando il file contenente le espressioni matematiche e un file in output, scriva sul file in output i risultati delle operazioni matematiche in **base-10**, uno per riga. Il programma deve anche controllare l'accuratezza dell'invocazione (e.g., numero di argomenti corretti) e la corretta apertura degli stream di input e output.

Il programma dovrà poter essere chiamato nel seguente modo:

```
./eserciziol.out input.txt output.txt
```

e dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
88
12
0
```

Note:

- Si assuma che le espressioni matematiche non contengano errori (e.g., codifiche errate, operazioni inesistenti, divisioni per zero, etc.). Si assuma anche che le espressioni matematiche siano separate da un carattere `\n`. Il numero massimo di cifre di ogni numero nel file di input è 10.
- Si ricorda che la base-19 è formata da numeri (0-9) e da lettere (A-I). Le lettere corrispondono ai valori che vanno da 10 a 16 ($A = 10, B = 11, \dots, I = 18$). La procedura di conversione è simile al caso binario, con la differenza della base b . Assumendo come esempio $e = 43G$, $decimale = \sum_{i=0} e[i] \cdot b^i$, dove $e[0] = G$, $e[1] = 3$ ed $e[2] = 4$.
- E' consentito l'utilizzo della funzione `int strlen (const char * str)` della libreria `<cstring>`.
- Non è consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- 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`.
- 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.

Information for graders:

Total of marks: 40