

Esame 20250221

Esercizio lode

(1) Esercizio Lode

Si consideri una coda che può contenere come elementi solamente valori interi corrispondenti ai caratteri dell'insieme `{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '+', '-', '*', '/', ' '}`, ossia le cifre da 0 a 9, gli operatori aritmetici di base e lo spazio.

Gli elementi nella coda rappresentano un'espressione aritmetica in cui l'ordine di precedenza degli operatori è solamente posizionale (NON valgono le regole tradizionali: semplicemente si devono valutare da sinistra a destra) e dove gli spazi NON sono significativi.

ESEMPIO (il simbolo `"."` di seguito rappresenta il carattere spazio per comodità!):

La coda contiene 39 elementi: `1..2.3+..4.5.-67*....8.+1.0..1./..9.-.1`

L'espressione si deve interpretare come $123+45-67*8+101/9-1$ (si devono ignorare tutti gli spazi) e i calcoli sono da eseguire semplicemente nell'ordine in cui si incontrano per cui il valore in questo caso è pari a 100 ($123+45=168$, poi $168-67=101$, poi $101*8=808$, poi $808+101=909$, poi $909/9=101$, poi $101-1=100$).

Una coda vuota si considera abbia come "valore" 0 (zero).

Implementare la funzione `"calc"` nel file `"calc.cc"` SENZA MODIFICARE ALCUN ALTRO FILE (neppure `"calc.h"`), avente segnatura `int calc();` in modo che calcoli il valore dell'espressione e modifichi la coda così da avere gli stessi dati ma senza tutti gli eventuali spazi intermedi attenendosi a quanto di seguito indicato:

- L'interazione con la coda deve avvenire solo ed esclusivamente con i metodi dedicati (descritti nel file `"queue.h"`) e non si può sfruttare la modalità implementativa!
- NON si può usare alcuna struttura di appoggio di alcun genere, ma solo variabili non strutturate dei seguenti tipi: `bool`, `int`, `char`, `int*`, `char*`, `int&`, `char&`, `string`!
- Nei calcoli usare sempre `"int"` ignorando eventuali arrotondamenti automatici (es.: $7/3$ restituisce il valore 2). Nel calcolare una divisione, se il divisore è 0, modificarlo in 1 (es.: $7/0$ è calcolato come $7/1$)! Non va cambiato nella coda!
- NON si possono definire né usare funzioni ricorsive!
- Si può assumere che l'espressione sia sempre ben definita (ossia la coda è vuota oppure contiene una rappresentazione di una sequenza che inizia con un operando e prosegue con uno o più coppie operatore+operando, non è possibile ci siano due operatori di fila o simboli non validi, mentre potrebbe esserci un solo operando).
- Si può compilare il progetto eseguendo `"make"` dentro la cartella (dov'è già presente un Makefile valido), altrimenti effettuare una compilazione manuale.

Note importanti.

- Scaricare il file `lode.zip` e scompattarlo in una cartella vuota.
- [`lode.zip`](#)
- Modificare solo il file `calc.cc` contenuto nell'archivio `lode.zip`.

- Caricare il solo file calc.cc per la valutazione.

ESEMPIO DI ESECUZIONE.

La funzione "main" è predisposta per richiamare la funzione "calc" con alcuni esempi statici e altri generati dinamicamente (in modo pseudo-casuale) e stampando poi per ogni caso: la coda originale preceduta dalla sua lunghezza, la coda modificata dopo l'invocazione di "calc" preceduta dalla sua lunghezza, un riscontro sul risultato (che mostra prima dei valori calcolato e atteso, due caratteri "T" o "F" relativi rispettivamente alla verifica che il valore calcolato sia corretto e che la coda sia stata modificata opportunamente).

```
make && ./calc && rm calc
g++ -o calc queue.h calc.h randexpr.h queue.cc calc.cc randexpr.cc main.cc
```

STATIC SAMPLES:

```
-----
(0)
(0)
[TT] 0 == 0

(2) : "74"
(2) : "74"
[TT] 74 == 74

(18) : "1 2 3 + 4 * 5      "
(7) : "123+4*5"
[TT] 635 == 635

(18) : "    20 + 7- 3/ 2   "
(8) : "20+7-3/2"
[TT] 12 == 12

(18) : "    1  0  /  0    "
(4) : "10/0"
[TT] 10 == 10
```

DYNAMIC SAMPLES:

```
-----
(61) : "4 + 84 8          / 2   65   * 3   6 7         +   4   1 0   "
(17) : "4+848/265*367+410"
[TT] 1511 == 1511

(162) : "3   9     1       /2     9     8     *      3   27     /     8     3     9       -4   0     5+   6
(39) : "391/298*327/839-405+655+144/579*621*306"
[TT] 0 == 0

(78) : "7   1   3       *3     7     4       * 7 7   1           /     2   4   9           +1   69"
(19) : "713*374*771/249+169"
[TT] 825857 == 825857

(87) : "9 38           + 6 7     4       * 7   3   9   +     5   3   1   +8   2   6   /8   8 5   "
(23) : "938+674*739+531+826/885"
[TT] 1347 == 1347

(63) : "3   2   9   /   2   5   4   +     6   2   5   /   5   5   6   "
(15) : "329/254+625/556"
[TT] 1 == 1
```

Information for graders:

Total of marks: 1