

Prova Finale di Algoritmi e Strutture Dati

A.A. 2019/2020

note generali

Introduzione

- Obiettivo: implementazione efficiente (e corretta!) di un algoritmo
- Logistica
 - Il codice sorgente sarà caricato su un server, compilato e fatto girare automaticamente
 - Scadenze (strette e vincolanti):
 - 9 luglio (ore 23.59) per i laureandi di luglio che ne faranno richiesta
 - 10 settembre (ore 23.59) per tutti gli altri
 - Tenete conto dei vincoli temporali per gestirvi!
 - Es. "inizio a pensarci a settembre" = fallimento quasi certo
- Esecuzione del progetto
 - implementazione nel linguaggio C
 - esclusivamente con libreria standard (libc)
 - no thread o tecniche di parallelizzazione

Valutazione

- Il programma deve compilare e girare correttamente
 - verranno resi disponibili dei casi di test come prova per controllare il corretto funzionamento del programma
 - dopo il caricamento sul server, il programma viene fatto girare su test, divisi in 2 parti: pubblici e privati
- si misura la correttezza (risultati in uscita) e l'efficienza (tempi di risposta e memoria occupata) del programma su vari casi di test
- dipendentemente dai risultati sui casi di test, potrete **calcolare** il voto
 - il voto è assegnato in modo **automatico** in base a come vanno i test
- non c'è recupero, ma il numero di “appelli” (= sottoposizioni) è praticamente illimitato

Sito per la sottoposizione del progetto

- <https://dum-e.deib.polimi.it/>
- Ogni studente avrà le proprie credenziali per accedere al sito
- Il sistema sarà presentato esaurientemente nel primo incontro coi tutor il 5 giugno alle 9.30

Operazioni disponibili sul sito

- scaricare le specifiche del progetto
- caricare (*submit / sottoponi*) il file da compilare e lanciare
 - si può usare il nome del file che si vuole, viene rinominato in automatico
 - dopo il caricamento, in automatico il file viene compilato e testato su un insieme di test “pubblici”
- lanciare (*play / usa*) i test “privati”, quelli su cui viene valutato il progetto

Plagi

- progetto da svolgere singolarmente ed in **totale autonomia** (no a gruppi)
- siete responsabili del vostro codice, quindi vi consigliamo fortemente di:
 1. Non caricarlo in repository **pubblici**
 2. Non passarlo per "ispirazione" a colleghi
- controllo plagi automatizzato
- in caso di copiatura **tutti i progetti coinvolti vengono annullati**

Scadenze

- Giovedì 10 settembre, ore 23.59
 - poi il sito per la sottomissione verrà chiuso
- Per i laureandi di luglio: giovedì 9 luglio, ore 23.59
 - avvisate (mandando un email al docente) che volete avere la valutazione a luglio

Ancora sul numero di appelli a disposizione

- C'è un solo appello? Solo due appelli se consideriamo la data per i laureandi?
 - NO!
 - Ce ne sono **infiniti**!
 - Potete fare tutti i tentativi che volete, fino a che non ottenete il voto che desiderate
 - Modificate il vostro codice
 - Lo sottoponete nuovamente
 - Verificate il nuovo punteggio ottenuto
- Ma dovete concludere il vostro lavoro entro la scadenza!
 - **NON** ci sarà recupero dopo quella data
 - ... quindi NON aspettate l'ultimo minuto per iniziare a lavorarci
 - Dopo l'incontro con i tutor, loggatevi subito e fate delle prove

edU, o ed con Undo multipli

Prova finale di Algoritmi e Strutture Dati AA 2019-20

Il tema

- Specifica completa all'indirizzo
<https://martinenghi.faculty.polimi.it/courses/api/ProvaFinale2020specifica.pdf>
- Un editor di testi con undo e redo, con i seguenti comandi:
 - ind_1, ind_2c cambia il testo alle righe comprese tra ind_1 ed ind_2 seguono $ind_2 - ind_1 + 1$ righe più una riga con il solo punto
 - ind_1, ind_2d cancella le righe tra ind_1 ed ind_2 (estremi inclusi)
 - ind_1, ind_2p stampa le righe tra ind_1 ed ind_2
 - nu fa l'undo degli ultimi n comandi (c o d)
 - nr fa il redo degli ultimi n comandi annullati con undo
 - q termina

Test di esecuzione

Input

1,2c
prima riga
seconda riga
.
2,3c
nuova seconda riga
terza riga
.
1,3p
1,1c
nuova prima riga
.
1,2p
2,2d
4,5p
1,3p
4,5d
1,4p
3u
1,6p
1r
1,3p
q

Testo

Output

Input

1,2c
prima riga
seconda riga

.

2,3c
nuova seconda riga
terza riga

.

1,3p
1,1c
nuova prima riga

.

1,2p
2,2d
4,5p
1,3p
4,5d
1,4p
3u
1,6p
1r
1,3p
q

Testo

prima riga
seconda riga

Output

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

prima riga

nuova seconda riga

terza riga

Output

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

prima riga

nuova seconda riga

terza riga

.

.

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

prima riga

nuova seconda riga

terza riga

.

.

.

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

prima riga

nuova seconda riga

terza riga

.

.

.

nuova prima riga

nuova seconda riga

terza riga

Input

1,2c

prima riga

seconda riga

.

2,3c

nuova seconda riga

terza riga

.

1,3p

1,1c

nuova prima riga

.

1,2p

2,2d

4,5p

1,3p

4,5d

1,4p

3u

1,6p

1r

1,3p

q

Testo

nuova prima riga

nuova seconda riga

terza riga

Output

prima riga

nuova seconda riga

terza riga

nuova prima riga

nuova seconda riga

.

.

nuova prima riga

terza riga

.

nuova prima riga

terza riga

.

.

prima riga

nuova seconda riga

terza riga

.

.

.

nuova prima riga

nuova seconda riga

terza riga

Breve tutorial del sito

Overview

- “Tutorial” è un semplicissimo problema, creato al solo scopo di sperimentare il funzionamento del sito
- Tutti gli altri sono i problemi da risolvere (i test che **devono** essere eseguiti con successo)

Overview

General information

The contest is currently running.

The contest started at May 1, 2020, 2:00:00 AM and will end at Oct 1, 2020, 2:00:00 AM.

You have an infinite number of tokens.

You can see the detailed result of a submission by using a token on it. Your score for each task will be the maximum among the tokened submissions and the last one.

Task overview

Task	Name	Time limit	Memory limit	Type	Files	Tokens
tutorial	Tutorial	1.000 second	256 MiB	Batch	batch[.c]	Yes
writeonly	writeonly	N/A	N/A	Batch	writeonly[.c]	Yes

“Tutorial”

- Problema: leggere da standard input 2 numeri interi separati da uno spazio; produrre su standard output la loro somma

Some details

Type	Batch	
Time limit	1 second	
Memory limit	256 MiB	
Compilation commands	C11 / gcc	<pre>/usr/bin/gcc -DEVAL -std=c11 -O2 -pipe -static -s -o Tutorial Tutorial.c -lm</pre>
Tokens	You have an infinite number of tokens for this task.	

Esempio di sottoposizione

- Creare e compilare il file `Tutorial.c`
 - Il sito suggerisce una riga di comando come questa:
`/usr/bin/gcc -DEVAL -std=gnu11 -O2 -pipe -static -s -o Tutorial Tutorial.c -lm`
- Verificare localmente il corretto funzionamento
- Caricare (“sottoporti”) il file `Tutorial.c` sul sito, e lanciare (“usa”) i test

Submit a solution

batch: no file selected

Previous submissions

Right now, you have infinite tokens available on this task. In the current situation, no more tokens will be generated.

Date and time	Status	Score	Files	Token
Jun 2, 2020, 5:16:42 PM	Evaluated details	1 / 1	<input type="button" value="Download"/>	<input type="button" value="Played"/>
May 29, 2020, 3:58:31 PM	Evaluated details	1 / 1	<input type="button" value="Download"/>	<input type="button" value="Played"/>
May 29, 2020, 8:18:12 AM	Evaluated details	1 / 1	<input type="button" value="Download"/>	<input type="button" value="Played"/>

Possibili strumenti utili

- valgrind (memory debugging, profiling)
 - <http://valgrind.org>
- gdb (debugging)
 - <https://www.gnu.org/software/gdb/>
- AddressSanitizer (ASan, memory error detector)
 - <https://github.com/google/sanitizers/wiki/AddressSanitizer>

Calcolo del voto

- Il task di tutorial non contribuisce in alcun modo all'esito della prova
- Sono previsti 6 task. Ognuno di questi si compone di:
 - 1 subtask pubblico - ovvero con input e output atteso pubblici (0 punti)
 - 3 subtask privati di difficoltà crescente (rispettivamente 3 + 1 + 1 punti)
- Il punteggio massimo ottenibile è 30 punti + 1 punto extra per la lode ottenibile superando un apposito task
- Per superare l'esame è necessario:
 - Superare almeno il subtask privato più semplice di ogni task
 - Non è sufficiente ottenere 18 punti se il vincolo di cui sopra non è soddisfatto
 - Verrà valutata l'ultima implementazione sottomessa in ogni task
 - L'implementazione valutata deve essere identica per tutti i task