



Piscina C

Rush 01

Sommario: Questo documento tratta il modulo Rush01 della Piscina C @ 42.

Indice

| | | |
|------------|-------------------|----------|
| I | Istruzioni | 2 |
| II | Preambolo | 3 |
| III | subject | 5 |
| IV | Annesso | 7 |

Capitolo I

Istruzioni

- Il gruppo SARÀ registrato alla difesa automaticamente.
- Non cancellate la registrazione, non ne avrete un'altra.
- Qualsiasi domanda riguardante questo modulo lo complicherà.
- Dovete seguire le procedure di consegna per tutti i vostri esercizi.
- Questo documento può subire variazioni fino a un ora prima della scadenza per la consegna.
- Moulinette compila per mezzo di `gcc` utilizzando queste flag: `-Wall -Wextra -Werror`.
- Se il vostro programma non compila, il voto sarà 0.
- Dovete completare il progetto con il team che vi è stato assegnato e presentarvi alla difesa con il team al completo
- Il progetto deve essere completato prima della difesa. Lo scopo di questa è quello di presentare e spiegare dettagliatamente il lavoro svolto.
- Ogni membro del gruppo deve essere a conoscenza di come funziona il progetto. Se decideste di dividere i compiti, assicuratevi che tutti abbiano compreso il lavoro svolto dagli altri. Durante la difesa vi saranno fatte varie domande e il voto finale sarà dato sulla base delle spiegazioni peggiori.
- Radunare il team sarà compito vostro. Avete molti mezzi a vostra disposizione per contattare i vostri teammates: telefono, email, piccioni viaggiatori, sedute spiritiche, etc. Non accetteremo scuse di alcun tipo. La vita non sempre è giusta.
- Detto questo, se avete davvero provato di tutto, ma uno dei vostri teammates rimane comunque irraggiungibile: completate comunque il progetto, e vedremo cosa possiamo fare durante la difesa. Anche se a mancare fosse il team leader avrete comunque accesso alla cartella di consegna.
- Non dovrebbe neanche esserci il bisogno di dirlo, ma il vostro progetto deve rispettare la Norma. Siate precisi.
- Enjoy !

Capitolo II

Preambolo

Here are some cool quotes from some random movies :

1. "Find a truly original idea. It is the only way I will ever distinguish myself. It is the only way I will ever matter."

-A Beautiful Mind

2. "You don't have to be the bad guy. You are the most talented, most interesting, and most extraordinary person in the universe. And you are capable of amazing things. Because you are the Special. And so am I. And so is everyone. The prophecy is made up, but it's also true. It's about all of us. Right now, it's about you. And you... still... can change everything."

-The Lego Movie

3. "Sometimes it is the people who no one imagines anything of who do the things that no one can imagine." -The Imitation Game

4. "There should be no boundaries to human endeavor. We are all different. However bad life may seem, there is always something you can do, and succeed at. While there's life, there is hope."

-The Theory of Everything

5. "Just because someone stumbles and loses their path, doesn't mean they're lost forever."

-X-Men Days of Future Past

6. "Where we're going we don't need roads"

-Back to the futur

7. "I'm bad, and that's good. I will never be good, and that's not bad. There's no one I'd rather be than me."

-Wreck-it Ralph


8. "KA-ME-HA-ME-HAAAAAAAAAAAA"

-Various movies

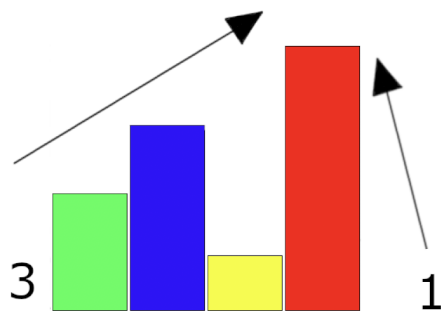
Movie culture won't help for this project even if it's important.

Capitolo III

subject

| | |
|---|--|
|  | Esercizio 00 |
| | Rush-01 |
| | Cartella per la consegna : <i>ex00/</i> |
| | File da consegnare : Tutti i file necessari |
| | Funzioni permesse : write, malloc, free |

- Il vostro codice sorgente sarà compilato così: `gcc -Wall -Wextra -Werror -o rush-01 *.c`
- La vostra cartella di consegna deve contenere tutti i file necessari a compilare il vostro programma.
- Create che risolva il problema seguente:
- Data una mappa 4x4, posizionare delle scatole di altezza da 1 a 4 su ogni casella disponibile in modo che ogni riga e colonna vedano il numero corretto di scatole da ogni possibile punto di vista (sinistra/destra per le righe, su/giù per le colonne).
- Ex: La scatola di altezza 3 nasconde la scatole di altezza 1 da sinistra, quindi abbiamo 3 scatole visibili, e solo una da destra, perché la scatola di altezza 4 nasconde le altre.



- Ognuna delle viste (2 per riga e 2 per colonna) avrà un valore prestabilito. Il vostro programma deve posizionare le scatole correttamente, assicurandosi che ogni riga ed ogni colonna contenga solo una scatola per tipo.
- Il vostro output deve contenere la prima soluzione trovata
- Ecco come eseguiremo il vostro programma :

```
> ./rush-01 "col1up col2up col3up col4up col1down col2down col3down col4down row1left row2left  
row3left row4left row1right row2right row3right row4right"
```

- (cf. annesso 1)
- "col1up" è il valore della vista verso l'alto della colonna di sinistra. Ognuno di questi rappresenta una stringa di caratteri dai valori compresi tra '1' e '4'.
- Questo è l'UNICO input che il vostro programma deve accettare. Ogni altro input deve essere considerato un errore.
- Ecco un esempio di input/output validi.

```
./rush-01 "4 3 2 1 1 2 2 2 4 3 2 1 1 2 2 2" | cat -e  
1 2 3 4$  
2 3 4 1$  
3 4 1 2$  
4 1 2 3$
```

- (cf. annessi 2 e 3)
- In caso di un errore o di soluzione non trovata stampate "Error" seguito da una nuova linea.

Capitolo IV

Annesso

Quello che segue è una rappresentazione artistica del vostro programma. Ovviamente dovete consegnare il programma descritto nel capitolo precedente.

Lo scopo di queste rappresentazione è puramente quello di aiutarvi a comprendere il progetto.

- Annesso 1:

| | col1up | col2up | col3up | col4up | |
|----------|----------|----------|----------|----------|-----------|
| row1left | | | | | row1right |
| row2left | | | | | row2right |
| row3left | | | | | row3right |
| row4left | | | | | row4right |
| | col1down | col2down | col3down | col4down | |

- Rappresentazione del vostro programma utilizzando col_up, col_down, row_left and row_right
- Annesso 2:

| | 4 | 3 | 2 | 1 | |
|---|---|---|---|---|---|
| 4 | | | | | 1 |
| 3 | | | | | 2 |
| 2 | | | | | 2 |
| 1 | | | | | 2 |
| | 1 | 2 | 2 | 2 | |

- Sostituendo col* e row*, otteniamo questo.
- Annesso 3:

| | | | | | |
|---|---|---|---|---|---|
| | 4 | 3 | 2 | 1 | |
| 4 | 1 | 2 | 3 | 4 | 1 |
| 3 | 2 | 3 | 4 | 1 | 2 |
| 2 | 3 | 4 | 1 | 2 | 2 |
| 1 | 4 | 1 | 2 | 3 | 2 |
| | 1 | 2 | 2 | 2 | |

- Il vostro programma deve riempire gli spazi vuoti utilizzando le regole stabilite nella prima parte.