



Piscina C

Rush 00

Sommario: Questo documento tratta il modulo Rush00 della Piscina C @ 42.

Indice

I	Istruzioni	2
II	Preambolo	4
III	Main subject	6
IV	Rush 00	7
V	Rush 01	9
VI	Rush 02	10
VII	Rush 03	11
VIII	Rush 04	12
IX	Consegna e valutazione tra pari	13

Capitolo I

Istruzioni

- Il gruppo verrà registrato automaticamente alla valutazione.
- Non cancellatela, non ve ne sarà assegnata 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.
- Gli esercizi sono presentati seguendo un ordine di difficoltà crescente. Ai fini della valutazione **NON** si prendono in considerazione gli esercizi se i precedenti non sono stati completati correttamente.
- Moulinette compila per mezzo di gcc utilizzando queste flag: -Wall -Wextra -Werror.
- Se il vostro programma non compila, il voto sarà 0.
- Gli esercizi dei Rush vanno completati in gruppi di 2, 3 o 4 persone.
- L'esercizio obbligatorio per il vostro team viene stabilito come segue:
Indice Alfabetico della prima lettera del login del team leader(1-26), modulo 5.
- 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 decidete 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.
- Se volete punti bonus potete consegnare altri esercizi oltre a quello obbligatorio.
- Moulinette non ha una mentalità aperta. Non proverà a comprendere il vostro codice se non rispetta la Norma. Moulinette utilizza un programma di nome **Norminator** per controllare la validità dei vostri file. TL;DR: sarebbe scocco tentare di consegnare un esercizio che non pass il controllo di **Norminator**.



Assicuratevi che l'esercizio obbligatorio del vostro gruppo funzioni perfettamente prima di considerare di completare esercizi bonus : se un esercizio bonus funziona, ma l'originale fallisce anche un solo test, otterrete uno 0.

Capitolo II

Preambolo

Here's the lyrics of a famous TV show for everyone :

[Verse 1]

I wanna be the very best
Like no one ever was
To catch them is my real test
To train them is my cause

I will travel across the land
Searching far and wide
Each pokemon to understand
The power that's inside

[Chorus]

Pokemon! Gotta catch 'em all! It's you and me
I know it's my destiny,
Pokemon! Oh you're my best friend
In a world we must defend
Pokemon! A heart so true
Our courage will pull us through,

You teach me and I'll teach you,
Pokemon! Gotta catch'em all

[Chorus]


Every challenge along the way
With courage I will face.
I will battle every day
To claim my rightful place.
Come with me,
The time is right,
There's no better team.
Arm in arm we'll win the fight!
It's always been our dream!

[Chorus]

I could bet you were singing right now, but it doesn't matter for the moment. And this subject is not related with Pocket Monster by the way...

Capitolo III

Main subject

	Esercizio 00
Rush0X	
Cartella per la consegna : <i>ex00/</i>	
File da consegnare : <code>main.c</code> , <code>ft_putchar.c</code> , <code>rush0X.c</code>	
Funzioni permesse : <code>write</code>	

- File da consegnare: `main.c`, `ft_putchar.c` e il vostro `rush0X.c`, dove '0X' rappresenta il numero dell'esercizio. Ad esempio `rush00.c`.
- Esempio di un `main.c` :

```
int main()
{
    rush(5, 5);
    return (0);
}
```

- Dovete quindi creare la funzione `rush` che prenda come argomenti due variabili di tipo `int`, chiamate, rispettivamente, `x` e `y`.
- La vostra funzione `rush` deve stampare a schermo un rettangolo di `x` caratteri di lunghezza per `y` caratteri di altezza.
- Il vostro `main` sarà modificato durante la difesa per controllare che abbiate gestito tutto quello che dovevate. Ecco uno dei test che effettueremo :

```
int main()
{
    rush(123, 42);
    return (0);
}
```

Capitolo IV

Rush 00

- `rush(5,3)` deve stampare :

```
$> ./a.out
o---o
|   |
o---o
$>
```

- `rush(5, 1)` deve stampare :

```
$> ./a.out
o---o
$>
```

- `rush(1, 1)` deve stampare :

```
$> ./a.out
o
$>
```

- `rush(1, 5)` deve stampare :

```
$> ./a.out
o
|
|
|
o
$>
```


- `rush(4, 4)` deve stampare :

```
$> ./a.out
o--o
| |
| |
o--o
$>
```

Capitolo V

Rush 01

- `rush(5,3)` deve stampare :

```
$> ./a.out
/***\
*   *
\***/
$>
```

- `rush(5, 1)` deve stampare :

```
$> ./a.out
/***\
$>
```

- `rush(1, 1)` deve stampare :

```
$> ./a.out
/
$>
```

- `rush(1, 5)` deve stampare :

```
$> ./a.out
/
*
*
*
\
$>
```

- `rush(4, 4)` deve stampare :

```
$> ./a.out
/**\
*  *
*  *
\**/
$>
```

Capitolo VI

Rush 02

- `rush(5,3)` deve stampare :

```
$> ./a.out
ABBBA
B  B
CBBBC
$>
```

- `rush(5, 1)` deve stampare :

```
$> ./a.out
ABBBA
$>
```

- `rush(1, 1)` deve stampare :

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deve stampare :

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` deve stampare :

```
$> ./a.out
ABBA
B  B
B  B
CBBC
$>
```

Capitolo VII

Rush 03

- `rush(5,3)` deve stampare :

```
$> ./a.out
ABBBC
B  B
ABBBC
$>
```

- `rush(5, 1)` deve stampare :

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` deve stampare :

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deve stampare :

```
$> ./a.out
A
B
B
B
A
A
$>
```

- `rush(4, 4)` deve stampare :

```
$> ./a.out
ABBC
B  B
B  B
ABBC
$>
```

Capitolo VIII

Rush 04

- `rush(5,3)` deve stampare :

```
$> ./a.out
ABBBC
B  B
CBBBA
$>
```

- `rush(5, 1)` deve stampare :

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` deve stampare :

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deve stampare :

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` deve stampare :

```
$> ./a.out
ABBC
B  B
B  B
CBBA
$>
```

Capitolo IX

Consegna e valutazione tra pari

Consegna gli esercizi nella tuo repository `Git` come al solito. Durante la difesa verrà considerato unicamente ciò che si trova all'interno della repository. Assicurati di controllare che i nomi dei tuoi file siano corretti.



Devi consegnare solo i file richiesti da questo documento.