

Verifica pipe: crypto

Il programma *crypto* contenuto nell'archivio allegato invia, **in forma cifrata**, una serie di frasi di Einstein della forma:

```
ALBERT EINSTEIN: L'uomo ha scoperto la bomba atomica, pero' nessun topo al mondo costruirebbe una trappola per topi.#
```

- Ogni frase inizia con "ALBERT EINSTEIN:" e termina con "#";
- La frase viene cifrata utilizzando una chiave random da 1 a 9 che viene sottratta al codice ASCII di ogni carattere. Ad esempio con la chiave 1 ALBERT diventa @KADQS (notare che @ precede A nella tabella ASCII).

Il programma **si aspetta in input la frase decifrata** e solo a quel punto ne propone una nuova della stessa forma ma **cifrata con una nuova chiave**. Solo se tutte le frasi vengono decifrate correttamente il programma stampa:

```
[*] Congratulazioni! Hai superato il crypto quiz!
```

Normalmente il programma legge e scrive su terminale. E' possibile specificare due pipe come argomenti da riga di comando usando rispettivamente le opzioni -i (input) e -o (output). In tale caso crypto **crea** le due pipe e le utilizza per comunicare. Alla fine dell'esecuzione le pipe vengono **eliminate**. Ad esempio:

```
./crypto -i /tmp/pipeIn -o /tmp/pipeOut
```

Il programma supporta altre opzioni:

- -v attiva la modalità verbosa.
- -h mostra le opzioni supportate.

Segue un esempio di esecuzione dove l'interazione avviene tramite standard input e standard output. In verde l'output in blu l'input e in rosso il messaggio di errore.

```
$ ./crypto
>I?BOQBFPQBKF7I$rj^kfq^$^so^$i^ploqb`ebp^mo^$jbofq^opf+
ALBERT EINSTEIN: L'umanita' avra' la sorte che sapra' meritarsi.#
<G=@MODINO@DI5gji_j"igjnoj`mnnj\g``i\djoo\m`)
ALBERT EINSTEIN: a caso
[ERRORE] letto a invece di I
```

Obiettivo:

L'obiettivo della verifica è di realizzare un programma che, usando le pipe, legga la frase cifrata (dalla pipe di output -o) e risponda con la frase in chiaro (sulla pipe di input -i). Il programma deve decifrare tutte le frasi proposte (cifrate con chiavi differenti) fino ad ottenere la stampa

```
[*] Congratulazioni! Hai superato il crypto quiz!
```

Suggerimenti:

- Il fatto che le frasi inizino allo stesso modo permette di **calcolare banalmente la chiave** (è sufficiente la prima lettera 'A!');
- Non è necessario *bufferizzare* la frase che leggete dalla pipe e costruire una stringa in quanto potete usare direttamente l'altra pipe come *buffer*;
- Ricordatevi che i **char** non sono altro che byte, quindi 'A'+1 è 'B';
- Il programma crypto **crea** e **distrugge** le pipe. Quando dovete testare la vostra soluzione, invocate **prima crypto poi il vostro programma**. Esempio:

```
./crypto -i /tmp/pipeIn -o /tmp/pipeOut & (sleep 1; ./soluzione)
```




- Se necessario, rendere eseguibile il programma utilizzando `chmod +x crypto`;

Nota Bene:

- Programmi che **non compilano** o **non superano il test** non verranno valutati (non consegnateli).
- Solo i programmi funzionanti **verranno valutati in base ai commenti**. Commentate in maniera appropriata il vostro programma e **inserite un commento iniziale** in cui spiegate l'idea risolutiva e specificate in che modo dobbiamo invocare il programma crypto per testare la vostra soluzione. Soluzioni non commentate non saranno valutate.
- **NON COPIATE E NON FATE COPIARE!** (Nel caso di soluzioni copiate verrà annullata la verifica di chi copia e chi ha fatto copiare indistintamente)

 [crypto.zip](#)

Stato consegna

Stato consegna	Consegnato per la valutazione
Stato valutazione	Valutata
Termine consegne	giovedì, 5 marzo 2020, 15:30
Tempo rimasto	Il compito è stato consegnato 13 min. 47 secondi in anticipo
Ultima modifica	giovedì, 5 marzo 2020, 15:16
Consegna file	  soluzione.c
Commenti alle consegne	 Commenti(0)

Commento

Valutazione	3,00 / 3,00
Data di valutazione	sabato, 14 marzo 2020, 15:16
Valutatore	 DILETTA OLLIARO
Commenti	

 [Annunci](#)

Vai a... 

[Verifica semafori: Imbuto 2.0](#) 

