ragazze DIGITALI

IDEE PER UN FUTURO SMART

Cosa faremo oggi

Costruiremo un programma che sarà in grado di convertire un normale testo in un codice segreto e viceversa.

Caesar Chiper

Vuoi criptare o decriptare un messaggio?
ciptare
Scrivi il tuo messaggio:
Questo è il corso Ragazze digitali, idee per un futuro smart
Inserisci la chiave (1-52)
13
Ecco il testo criptato:
dHrFGB zèz zvy zpBEFB zentnMMr zqvtvGnyv,z zvqrr zCrE zHA zsHGHEB zFznEG

Outline

Crittografia 1/3

Vediamo di imparare qualche nozione elementare di crittografia che ci può essere utile per scrivere il nostro programma

Per iniziare

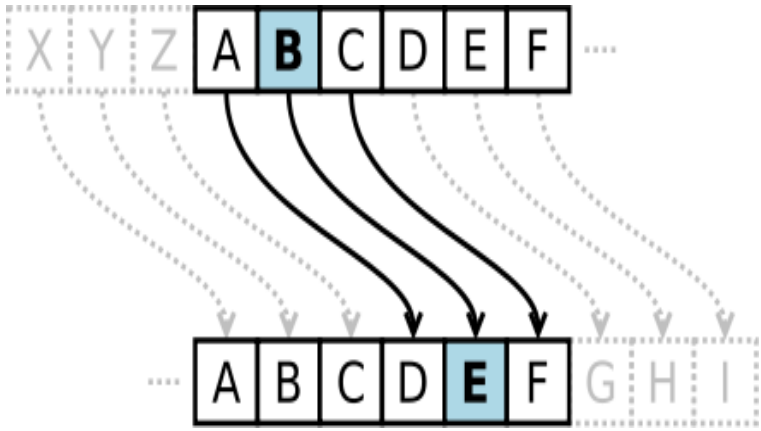
- **chiper** = è il sistema, l'insieme delle regole secondo le quali *criptiamo* un messaggio
- **plaintext** = testo che vogliamo nascondere e mantenere segreto
- **chipertext** = testo trasformato
- un *plaintext* viene **criptato** in un *chipertext*
- un *chipertext* viene **decriptato** in un *plaintext*
- **chiave** = valore segreto con il quale si decripta un messaggio criptato usando un determinato **chiper**

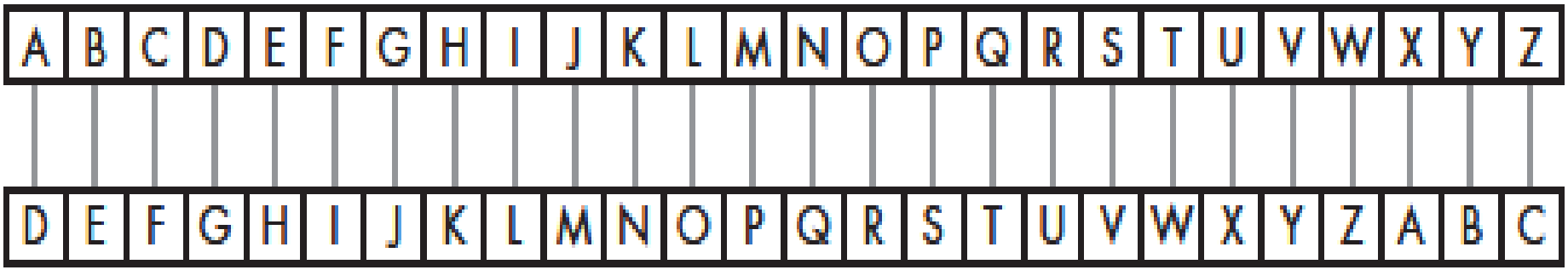
Crittografia 2/3

Esistono tantissimi chiper, sistemi per crittografare un messaggio, ciascuno con la propria chiave. Per costruire il nostro programma ci interesseremo del **Caeser chiper**, ovvero del *cifrario di Cesare*.. Si, proprio quel Cesare che pensate! ...E' parecchio vecchio come chiper, ma tutt'ora ancora perfettamente funzionante.

Crittografia 3/3

Con questo Caeser chiper, i messaggi vengono criptati rimpiazzando ciascuna lettera con una lettera *"shiftata"*, ovvero spostata.
Per esempio, se *shiftiamo* di 3 lettere, avremo che la lettera **B** diventerà una **E**, e così via. Per decriptare i messaggi, verranno shiftate indietro le lettere e quindi una **E** diventerà una **B**.





La **chiave** del Caeser chiper è il numero di lettere shiftate.

Outline

string.find()

Come funziona

- E' un metodo che restituisce la posizione in cui si trova la stringa passata al metodo rispetto alla stringa su cui è invocato il metodo.
- Proviamo a vedere con degli esempi come funziona

```
>'Hello world'.find('H')
0
>'Hello world'.find('o')
4
>'Hello world'.find('ell')
1
>'Hello world'.find('xyz')
-1
```

- La numerazione degli indici parte da 0 e non da 1!
- Di 'o' ce ne sono due, ma viene restituito l'indice della prima occorrenza trovata
- Se si ricerca una stringa, viene restituito l'indice dell'inizio della stringa
- Se si cerca una stringa non presente, viene restituito -1

Outline

len()

Come funziona

- E' un metodo che restituisce il numero di caratteri presenti nella stringa passata in input
- Proviamo a vedere con degli esempi come funziona

```
SYMBOLS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
MAX_KEY_SIZE = len(SYMBOLS)
```

- In questo esempio, la variabile globale `SYMBOLS` contiene una stringa rappresentante tutte le lettere dell'alfabeto.
- La variabile globale `MAX_KEY_SIZE`, invece, contiene il risultato della chiamata alla funzione **len()** invocata con parametro la stringa di cui vogliamo calcolare la lunghezza

Outline

Caeser chiper 1/5

Ora tocca a voi!

- Definisci innanzitutto due variabili globali, una contenente tutte le lettere dell'alfabeto (minuscole e maiuscole) e un'altra che contenga il numero delle lettere definite in precedenza
- Costruisci una funzione che chieda all'utente se vuole criptare o decriptare un messaggio e che restituisca la modalità scelta dall'utente; altrimenti, se l'utente inserisce un carattere o una stringa non inerente alla scelta, viene mostrato un messaggio di spiegazione su cosa bisogna inserire e viene riproposta la domanda iniziale.
- **Suggerimento**: per controllare cosa inserisce l'utente, può essere di aiuto convertire l'input in caratteri *lowercase*

Caeser chiper 2/5

Ora tocca a voi!

- Costruisci una funzione che chieda all'utente di inserire il messaggio che vuole criptare/decriptare e lo restituisca come valore di ritorno della funzione.
- Costruisci una funzione che chieda all'utente di inserire la chiave di cifratura, che sarà un numero compreso tra 1 e il numero delle lettere definite all'inizio. Controllare che il valore inserito sia all'interno di questo range. Nel caso non lo fosse, deve essere richiesto all'utente di inserire la chiave; se è dentro al range, viene restituito dalla funzione.

Caeser chiper 3/5

Ora tocca a voi!

- Costruisci una funzione che effettivamente cripta o decripta il messaggio in base a cosa è stato scelto dall'utente e alla chiave scelta.
- Per ogni simbolo (carattere) del nostro messaggio, se il carattere è presente nella nostra lista caratteri (ovvero vuol dire che appartiene all'alfabeto), dobbiamo salvarci l'indice del nuovo carattere da sostituire, in base alla chiave scelta.
- Una volta trovato il nuovo indice, creiamo un array con la nuova stringa, inserendo carattere per carattere. Infine si restituisca l'array.
- Se il carattere non è presente nella nostra lista, copiamo semplicemente il vecchio carattere senza sostituirlo.

Caeser chiper 4/5

Ora tocca a voi!

- In questa ultima funzione per prima cosa bisogna verificare se è stata scelta la modalità decriptazione: con essa, infatti, è necessario rendere negativa la chiave, così che nella fase di sostituzione del carattere esso venga sostituito con il corrispettivo simbolo.
- Inseriamo quindi questo pezzetto di codice:

```
if mode[0] == 'd':
    key = -key
translated = '' # inizializziamo a nullo il
   vettore che conterra' la stringa finale
```

Caeser chiper 5/5

Ora tocca a voi!

- Come ultima cosa richiamate le funzioni, salvate i valori restituiti e stampate la stringa criptata o decriptata!
- Ed ecco, il nostro cifrario è completato!