Lavorare con le stringhe in *Mathematica* Cryptografia a sostituzione.

Funzioni base

Una stringa e' una lista di caratteri rappresentati con i doppi apici: "ciao, mondo!" e' una stringa.
Per avere la lunghezza di una stringa c'e' la funzione StringLength[s]. Per estrarre il carattere nella posizione n-esima, si usa la funzione StringTake[s,{n}]. Esempi:

```
s = "ciao, mondo!";
StringLength[s]
StringTake[s, {7}]
```

```
12
```

```
m
```

Per concatenare due stringhe s1 ed s2 c'e' la funzione StringJoin[s1,s2], abbreviabile in s1<>s2. Es:

```
"ciao" <> " mondo"
```

```
ciao mondo
```

Per mettere in maiuscolo una stringa c'e' ToUpperCase[s]; per mettere in minuscolo ToLowerCase[s].

Conversione ASCII

Per convertire un carattere nel suo equivalente numerico ASCII e viceversa ci sono due funzioni (una inversa dell'altra): ToCharacterCode e FromCharacterCoden (vedi help on-line)

Definiamo due funzioni che fanno una cosa analoga, ma mappando i caratteri nell'intervallo [0,25]:
in questo modo faremo corrispondere 0->"A" e 25->"Z".

```
a2c[ascii_] := FromCharacterCode[ascii + 65]
c2a[char_] := First[  ToCharacterCode[char]  ] - 65
```

Esempio:

```
a2c[0]
c2a["A"]
```

```
A
```

```
0
```

Come si vede, e' stato sufficiente sommare o sottrarre 65 (che corrisponde in ascii alla posizione del carattere A) per ottenere il risultato.

Criptografia a sostituzione

Si tratta di una funzione crypt[M,K] che ha in input un messaggio M (una stringa) e una chiave K (di vari caratteri) e che produce in output una stringa della stessa lunghezza ma con tutti i caratteri ruotati ciclicamente, in base alla chiave K. Per semplicita', considereremo validi solo i caratteri A-Z (solo maiuscolo, niente spazi, punteggiatura, etc.)

```
crypt[M_, K_, op_] := Module[ {mesg, key, k, i, j = 0, old, new, ris},
   key = ToUpperCase[K]; mesg = ToUpperCase[M];
    ris = "";
   For[ i = 1, i ≤ StringLength[mesg], i = i + 1,
    k = c2a[StringTake[key, { Mod[j, StringLength[key]] + 1}]];
    old = c2a[ StringTake[mesg, {i}] ];
    new = Mod[ old + op * k, 26];
    ris = ris <> a2c[new];
    j = j + 1;
   ];
   ris
 ]
```

Il parametro (op) puo' valere +1 (codifica) o -1 (decodifica).

Test

```
chiaro = "nelmezzodelcammindinostravita";
```

```
criptato = crypt[chiaro, "Benigni", 1]
```

```
OIYUKMHPHRTINUNMALOAWTXEIBVBB
```

```
crypt[criptato, "Benigni", -1]
```

```
NELMEZZODELCAMMINDINOSTRAVITA
```

Commenti

1) Notare la presenza di due indici (i) e (j): nel mentre che (i) "spazzola" tutto il messaggio, (j) percorre ripetutamente la chiave. Se la chiave, ad esempio, e' lunga 6 il trucco viene realizzato contanto modulo 6, e sommando 1 al resto modulare: Mod[j,6]+1 infatti "ruota" tra 1 e 6, cioe' tra il primo e l'ultimo carattere nella stringa KEY.
2) Notare come il messaggio criptato venga creato accodando ogni nuovo carattere nella variabile RIS, inizialmente vuota.
3) Notare il modo in cui viene creato il codice ascii del carattere criptato NEW, a partire dal codice ascii del carattere in chiaro OLD: sommando il codice della chiave prendendo il modulo 26. Questo garantisce che i caratteri oltre la Z si "arrotolino" sulla A.