



Conversione testo in ottetti per messaggi SMS (prima parte)


http://www.vbsimple.net/activity/act_02.htm

Richiesta di: Marcello Gaidone - 6 Novembre 2000

Difficoltà: **5 / 5**

*Devo trasformare (ad esempio) la parola **hellohello** in una stringa formata da 8 bit, in modo da poterla utilizzare per mandare SMS sui cellulari.*

Ammetto che questa richiesta mi ha impegnato per almeno un'ora (e un'altra ora aggiuntiva per effettuare alcune correzioni), ma alla fine siamo riusciti a produrre i risultati voluti. Per effettuare quest'operazione è necessario conoscere bene i diversi sistemi di numerazione, in particolare la conversione da sistema decimale a sistema binario, per questo motivo le due funzioni di conversioni da un sistema all'altro non verranno approfondite. Per effettuare queste operazioni consultare la sezione [Informazioni aggiuntive](#).

Prima di cominciare dobbiamo prendere coscienza che l'invio di messaggi telefonici può avvenire su qualsiasi piattaforma. Pertanto l'alfabeto da utilizzare non è il solito charset [ASCII](#). Esiste un alfabeto apposito che definisce i codici per tutti i simboli utilizzabili in un messaggio SMS. Inseriremo l'intero alfabeto e le formule per trattarlo in un modulo  separato. Purtroppo alcuni simboli non possono essere rappresentati con il charset ASCII e pertanto non saranno convertiti; questi simboli sono alcune lettere greche maiuscole. Ulteriori informazioni su questo alfabeto si trovano all'indirizzo <http://www.dreamfabric.com/sms/>.

L'alfabeto dedicato utilizza una cifra composta da sette bit per la maggior parte dei caratteri. Soltanto 10 caratteri non rientrano in tale range. I caratteri che non rientrano in tale spazio allocato saranno trattati come una coppia di tali valori, pertanto saranno rappresentati da due caratteri.

All'interno del primo modulo definiamo le seguenti variabili globali pubbliche:

```
1. Private LettereAlfabeto(127) As String
2. Private ExtraAlfabeto(9) As String
3. Private ValoreExtraAlfabeto(9) As Integer
4.
```

Alla riga 1 abbiamo definito un [array](#) di 128 (ovvero 7 cifre binarie) stringhe compresa quella a posizione zero di nome **LettereAlfabeto**. Esso conterrà tutti i caratteri normali composti da un solo carattere.

Alla riga 2 abbiamo un altro array di 10 stringhe di nome **ExtraAlfabeto** per contenere i dieci simboli che sono composti da due caratteri. Ad esso si affianca un terzo array di nome **ValoreExtraAlfabeto**, sempre di 10 elementi, che conterrà il codice corrispondente

al simbolo speciale.

Tali vettori andranno riempiti (inizializzati) da una funzione apposita. Il suo nome è **InizializzaAlfabeto**, dichiarata pubblica. Ne vediamo un pezzetto perché è molto ampia:

```
5. Public Sub InizializzaAlfabeto()
6.     LettereAlfabeto(0) = "@"
7.     LettereAlfabeto(1) = "£"
8.     LettereAlfabeto(2) = "$"
```

Seguono i vari simboli...

```
144.     LettereAlfabeto(126) = "ü"
145.     LettereAlfabeto(127) = "à"
146.
147.     ExtraAlfabeto(0) = vbFormFeed: ValoreExtraAlfabeto(0) = 6922
148.     ExtraAlfabeto(1) = "^": ValoreExtraAlfabeto(1) = 6932
149.     ExtraAlfabeto(2) = "{": ValoreExtraAlfabeto(2) = 6952
150.     ExtraAlfabeto(3) = "}": ValoreExtraAlfabeto(3) = 6953
151.     ExtraAlfabeto(4) = "\": ValoreExtraAlfabeto(4) = 6959
152.     ExtraAlfabeto(5) = "[": ValoreExtraAlfabeto(5) = 6972
153.     ExtraAlfabeto(6) = "~": ValoreExtraAlfabeto(6) = 6973
154.     ExtraAlfabeto(7) = "]": ValoreExtraAlfabeto(7) = 6974
155.     ExtraAlfabeto(8) = "|": ValoreExtraAlfabeto(8) = 6976
156.     ExtraAlfabeto(9) = "•": ValoreExtraAlfabeto(9) = 7013
157. End Sub
```

Nelle righe 6-145 abbiamo i 128 simboli standard composti da un solo numero. Il loro numero è la posizione all'interno dell'array.

Alle righe 147-156 abbiamo i dieci simboli speciali, composti da due caratteri. Il valore che viene inserito negli elementi di **ValoreExtraAlfabeto** sono dati dalla concatenazione [esadecimale](#) dei due simboli separati.

Ad esempio il simbolo "^" si compone dei due caratteri a cui corrispondono i valori esadecimali 1B e 14 ovvero i numeri decimali 27 e 20. Concatenando i due valori esadecimali otteniamo il numero esadecimale 1B14 ovvero il valore decimale 6932, che abbiamo inserito nell'array **ValoreExtraAlfabeto**.

Questo genere di operazione macchinosa si è resa necessaria per poter distinguere i due bytes di cui si compone ogni simbolo speciale.

Il modulo si completa con la funzione che si occupa di restituire il codice cui corrisponde ogni simbolo, speciale o non.

```
159. Public Function TrovaCarattereAlfabeto7(ByVal Carattere As String) As Integer
160.     Dim Conta As Integer
161.
162.     For Conta = LBound(LettereAlfabeto) To UBound(LettereAlfabeto)
163.         If LettereAlfabeto(Conta) = Carattere Then Exit For
164.     Next Conta
165.     If Conta > UBound(LettereAlfabeto) Then
166.         For Conta = LBound(ExtraAlfabeto) To UBound(ExtraAlfabeto)
167.             If ExtraAlfabeto(Conta) = Carattere Then Exit For
168.         Next Conta
169.         If Conta <= UBound(ExtraAlfabeto) Then
170.             Conta = ValoreExtraAlfabeto(Conta)
171.         Else
172.             Conta = 0
173.         End If
174.     End If
```

```
174.      End If
175.      TrovaCarattereAlfabeto7 = Conta
176. End Function
```

La funzione è denominata **TrovaCarattereAlfabeto7**, richiede una stringa (il carattere da cercare) e riporta un numero intero corrispondente al codice nell'alfabeto appena visto.

Alle righe 162-164 effettuiamo un primo controllo dell'alfabeto standard (i 128 elementi) alla ricerca del carattere di cui trovare il codice. Se il codice viene trovato viene forzata l'uscita dal ciclo for (riga 163). Pertanto, l'uscita dal ciclo for indicherà che abbiamo trovato il carattere richiesto oppure che dopo l'intera analisi dell'array **LettereAlfabeto** il carattere non è stato trovato. In questo ultimo caso però, il valore di Conta sarà 128 (ovvero un numero superiore alla dimensione dell'array).

Sfrutteremo pertanto proprio tale valore. Se il valore di **Conta** è inferiore o uguale a 127, il valore **Conta** sarà il numero che dovremo restituire in uscita alla funzione. Nel caso contrario dovremo controllare il secondo array con i simboli speciali.

Pertanto alle righe 166-168 effettueremo questo secondo controllo. All'uscita del ciclo Conta avrà il valore corrispondente alla posizione del carattere speciale nell'array. Ecco perchè alla riga 170 assegniamo a Conta il numero corrispondente nell'array **ValoreExtraAlfabeto**. Nel caso che il carattere non fosse stato trovato neanche nel secondo alfabeto, cancelleremo il carattere impostando per esso il valore 0.

La funzione si chiude con la restituzione del numero contenuto in **Conta**.

La soluzione, purtroppo è molto macchinosa, ma l'alfabeto da utilizzare è molto diverso dall'ASCII utilizzabile sui normali PC.

[Segue parte 2 >>](#)

[Fibia FBI](#)

7 Novembre 2000

Corretto il 28 Marzo 2001



[Torna all'introduzione delle Richieste dei lettori](#)
