

## 1. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti da tastiera un numero positivo di 4 cifre decimali e che visualizzi sullo schermo:

- Il numero stesso in codice binario.
- Il suo quadrato in formato decimale
- Il suo quadrato in codice binario.
- Il suo cubo in formato decimale.
- Il suo cubo in codice binario

## 2. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti da tastiera un numero decimali compreso fra -999 e +999 e che visualizzi sullo schermo:

- Il numero stesso in complemento a due.
- Il suo quadrato in formato decimale
- Il suo quadrato in complemento a due..

## 3. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti da tastiera un numero di 3 cifre esadecimale (intero in complemento a due) e che visualizzi sullo schermo:

- Il numero in formato decimale

Si consideri poi lo stesso numero esadecimale come frazione binaria in complemento a due e si visualizzi il valore del numero in formato decimale.

## 4. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una sequenza di numeri (max 60) e la visualizzi sullo schermo.
- Accetti una sequenza di quattro numeri e la visualizzi sullo schermo.
- Verifichi quante volte la sequenza dei quattro numeri compare nella prima sequenza e stampi il risultato.
- Verifichi quante volte la sequenza appare con un errore e stampi il risultato.

## 5. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una frase che contenga solo caratteri alfabetici (max 60) e la visualizzi sullo schermo.
- Accetti una sequenza di quattro caratteri alfabetici e la visualizzi sullo schermo.
- Verifichi quante volte la sequenza dei quattro caratteri alfabetici compare nella prima sequenza e stampi il risultato.
- Verifichi quante volte la sequenza appare con un errore e stampi il risultato.

## 6. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una parola di 32 bit nella sua rappresentazione esadecimale.
- Visualizzi sullo schermo la rappresentazione esadecimale delle 32 configurazioni che si ottengono facendo circolare il dato precedente in un registro di 32 posizioni richiuso su se stesso.
- Accanto ad ogni rappresentazione si indichi il valore in formato decimale del numero considerato come frazione binaria in complemento a due.

## 7. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una parola di 32 bit nella sua rappresentazione esadecimale.
- Visualizzi sullo schermo la rappresentazione esadecimale delle 32 configurazioni che si ottengono facendo circolare il dato precedente in un registro di 32 posizioni richiuso su se stesso.
- Accanto ad ogni rappresentazione si indichi il valore in formato decimale del numero considerato come numero binario intero in complemento a due.

## 8. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una sequenza di numeri decimali positivi e negativi di 4 cifre più segno.
- Stampi su schermo la lista di numeri inseriti in ordine crescente.

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire un numero,
  - stampare la lista ordinata dei numeri già inseriti,
  - uscire dal programma.
3. Ritornare, se del caso, al punto 2.

## 9. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una sequenza di tre cifre esadecimale che rappresentano numeri interi in complemento a due positivi e negativi.
- Stampi su schermo la lista di numeri inseriti in ordine crescente con accanto il valore decimale del numero con segno.

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire un numero,
  - stampare la lista ordinata dei numeri già inseriti,
  - uscire dal programma.
3. Ritornare, se del caso, al punto 2.

## 10. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera parole (da 1 a 10 caratteri).
- Stampi su schermo la lista delle parole inserite in ordine di lunghezza.

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire una parola,
  - visualizzi la lista ordinata delle parole già inserite,
  - uscire dal programma.
3. Ritornare, se del caso, al punto 2

Al termine dell'inserimento e della visualizzazione il programma deve ripetere le operazioni previste al punto 2.

## 11. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera parole (da 1 a 10 caratteri).
- Stampi su schermo la lista delle parole inserite in ordine alfabetico.

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire una parola,
  - visualizzi la lista ordinata delle parole già inserite,
  - uscire dal programma.
3. Ritornare, se del caso, al punto 2

Al termine dell'inserimento e della visualizzazione il programma deve ripetere le operazioni previste al punto 2.

## 12. CALCOLATORI ELETTRONICI X

Scrivere un programma in assembler che agisca da semplice guida telefonica:

- Accetti da tastiera il cognome di un utente telefonico (max 15 campi alfanumerici)
- Il numero di telefono (max 12 caratteri numerici).

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire un nuovo utente,
  - stampare la lista degli utenti inseriti con i rispettivi numeri di telefono.
  - modificare il numero di telefono di un utente.
  - uscire dal programma.

Al termine dell'inserimento e della richiesta il programma deve ripetere le operazioni previste al punto 2.

### 13. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che agisca da semplice guida telefonica:

- Accetti da tastiera il cognome di un utente (max 15 campi alfanumerici)
- L'indirizzo di posta elettronica dell'utente.

Il programma deve operare nel seguente modo:

1. Presentarsi.
2. Richiedere se si vuole:
  - inserire un nuovo utente,
  - stampare la lista degli utenti inseriti con i rispettivi indirizzi di posta elettronica.
  - modificare l'indirizzo di posta elettronica di un utente.
  - uscire dal programma.

Al termine dell'inserimento e della richiesta il programma deve ripetere le operazioni previste al punto 2.

### 14. CALCOLATORI ELETTRONICI X

Scrivere un programma in assembler in grado di gestire un'agenda giornaliera. L'agenda deve memorizzare gli impegni orari di una giornata di lavoro dalle ore 9.00 alle ore 17.00.

Il programma deve:

1. Presentarsi visualizzando gli impegni nel formato seguente:  
9.00 .....  
10.00 .....  
11.00 .....  
.....  
17.00 .....
2. Permettere l'inserimento di nuovi impegni e al termine la visualizzazione degli stessi.
3. Permettere la cancellazione di tutti gli impegni.
4. Permettere di uscire dal programma.

L'impegno orario è rappresentato da una stringa alfanumerica (max 50 caratteri).

## 15. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che esegua il prodotto di due matrici A e B e dia come risultato la matrice C. Il programma deve:

- Accettare da tastiera le due matrici (gli elementi delle matrici A e B, di dimensione 2x2, sono numeri positivi inferiori a 10.)
- Presentare su video la matrice C risultato del prodotto.

Scrivere successivamente lo stesso programma nel caso di dimensione 3x3.

## 16. CALCOLATORI ELETTRONICI

Scrivere un programma assembler che calcoli i primi 400 valori della sequenza generata con la seguente legge:

$$X_{i+1} = |aX_i + b|_{\text{mod}256}$$

dove a e b sono numeri compresi fra 0 e 255.

Il programma deve:

- Acquisire i valori di a e b e quello iniziale di  $X_i$  ;
- Visualizzare la sequenza su 20 righe di 20 numeri con il seguente formato (b rappresenta uno spazio):

bbb1bb25b145bb30b100bbb0...

- Verificare dopo quante iterazioni la X riassume il valore iniziale.

## 17. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che, dato un numero compreso fra 0 e 99, calcoli tutte le potenze dello stesso fino a raggiungere il numero massimo contenuto in 32 bit.

Il programma deve:

- Accettare da tastiera il numero.
- Presentare su video la serie di potenze significative.

## 18. CALCOLATORI ELETTRONICI

Scrivere un programma assembler che accetti un numero decimale in base 10 con il seguente formato: zz,zzz e ne calcoli la trasformazione in base 2 con il formato xxxxx,xxxxx. La trasformazione è ovviamente approssimata.

Il programma deve stampare come output:

- La trasformazione in base 2 nel xxxxx,xxxxx.
- La ritrasformazione in base 10 nel formato zz,zzz del numero in base 2 calcolato al punto precedente.

## 19. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che, dato un numero compreso fra 0 e 9999, lo scomponga nei suoi fattori primi. Il programma deve:

- Accettare da tastiera il numero.
- Presentare su video la serie dei suoi fattori primi con la relativa molteplicità.

Esempio:  $150 = 2 \cdot 3 \cdot 5(2)$

## 20. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che esegua la trasformazione fra numeri in basi diverse. In particolare il programma deve:

1. Accettare da tastiera un numero K che rappresenti la base della numerazione del primo numero.
2. Accettare da tastiera un numero N nella base prima indicata.
3. Accettare da tastiera un numero L che rappresenti la base di numerazione nella quale occorre trasformare il numero N.

Il programma deve fornire il numero specificato ai punti 1) e 2) nella base indicata in 3).

Sono da considerare i seguenti limiti:

- $2 < K < 20$ .
- Il valore del numero rappresentato da N (in base K) deve essere minore di 65536 (decimale) =  $2^{16}$ ; in caso contrario il programma deve segnalare errore.
- I simboli da usare per K sono:  $2 \rightarrow 2; \dots; 9 \rightarrow 9; 10 \rightarrow A; \dots; 20 \rightarrow M$ :
- Gli stessi simboli devono essere usati per rappresentare i numeri.

Esempio:  $23B$  (base C) =  $335$  (base A).

## 21. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che, dato un numero compreso fra 0 e 64000, ne calcoli la radice quadrata e la radice cubica con approssimazione all'intero inferiore. Il programma deve:

- Accettare da tastiera il numero.
- Presentare su video:
  - la radice quadrata e il quadrato approssimato.
  - la radice cubica e il cubo approssimato.

Esempio:

numero: 1612

radice quadrata: 40 1600

radice cubica: 11 1331

## 22. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera 10 numeri compresi tra -9999 e 9999.

- Li ordini in modo crescente.
- Li visualizzi sullo schermo.

### 23. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che muova di una posizione un cursore (qualsiasi) sullo schermo nelle quattro direzioni (nord, sud, est, ovest) utilizzando come comandi quattro caratteri diversi definiti a priori.

Il programma deve inoltre:

- partire con il cursore circa al centro dello schermo,
- quando il carattere raggiunge uno dei bordi impedire che esca dallo schermo.

Nel loop principale del programma è opportuno inserire un ritardo; considerare la posizione più opportuna nel loop per il ritardo stesso. In una prima fase realizzare il programma indicato non considerando controlli sui bordi dello schermo.

### 24. CALCOLATORI ELETTRONICI X

Scrivere un programma in assembler che muova automaticamente un cursore (qualsiasi) sullo schermo in una delle quattro direzioni (nord, sud, est, ovest). Utilizzare quattro tasti qualsiasi per fare cambiare la direzione di spostamento del cursore.

Il programma deve inoltre:

- partire con il cursore circa al centro dello schermo,
- considerare lo schermo una struttura toroidale.

Nel loop principale del programma è opportuno inserire un ritardo; considerare la posizione più opportuna nel loop per il ritardo stesso.

In una prima fase realizzare il programma indicato non considerando controlli sui bordi dello schermo.

### 25. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che disegni sullo schermo un rettangolo centrato sullo schermo e di altezza e base definita.

Il programma deve svolgere alcune funzioni, quando viene premuto un tasto:

- Tasto “+” è aumentata l’altezza del rettangolo disegnato.
- Tasto “-” è diminuita l’altezza del rettangolo.
- Tasto “1” è aumentata la base
- Tasto “2” è diminuita la base.
- Tasto “\$” si esce dal programma.

Contemporaneamente alla visualizzazione del rettangolo in basso a destra sullo schermo si stampi “area” del rettangolo intesa come prodotto del numero delle righe per il numero delle colonne racchiuse nel rettangolo.

### 26. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti in ingresso una frase con un massimo di 500 caratteri e che di questa calcoli:

- il numero di parole (si considerano separatori: lo spazio, i segni di punteggiatura, gli apostrofi).
- la lunghezza massima e la lunghezza minima delle parole.
- la lunghezza media delle parole (questo valore deve essere calcolato con una cifra decimale significativa).

## 27. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti in ingresso una frase con un massimo di 500 caratteri e che successivamente:

- chieda all'utilizzatore di indicare un carattere o una sequenza di caratteri (max 4 caratteri).
- risponda indicando tutte le parole nelle quali compare il carattere o la sequenza di caratteri precedentemente indicata.
- si considerino separatori fra parole gli spazi e i segni di punteggiatura: punto, virgola, due punti e punto e virgola.

## 28. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso un numero a una cifra.
- Visualizzi il numero sullo schermo con un'altezza di circa 5 cm in una schematizzazione a 7 segmenti.

Ripetere il secondo punto considerando:

- Numeri a 2 cifre.
- Numeri con un numero di cifre qualsiasi compreso fra 1 e 4.

## 29. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una stringa contenente al massimo 80 caratteri.
- Accetti da tastiera una stringa contenente al massimo 4 caratteri.
- Determini e visualizzi il numero di ricorrenze della seconda stringa nella prima stringa.

In una seconda fase si consideri la seconda stringa formata di esattamente 4 caratteri, e si determino i numeri di ricorrenze parziali formate da solo 3 caratteri e 2 caratteri nell'ordine stabilito dalla seconda stringa (matching approssimati a 3 o 2 caratteri).

## 30. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti da tastiera una stringa contenente 8 caratteri ASCII.
- Visualizzi tale stringa in formato esadecimale.



- Aggiunga un bit di parità ad ogni carattere della stringa ASCII, in modo che ogni carattere sia rappresentato da 9 bit e con un numero pari di 1.
- Stampi la stringa di bit risultante (72 bit = 8 caratteri da 9 bit) in formato esadecimale.

### 31. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che consenta di gestire un'agenda giornaliera di impegni. La giornata è suddivisa in ore dalle 8.00 alle 20.00. Il programma deve consentire di:

- Visualizzare gli impegni sotto forma di tabella. Ad ogni ora viene associata una riga con due campi per specificare l'ora e il tipo di impegno.
- Modificare e/o inserire nuovi impegni per ogni singola ora della giornata.

### 32. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti in ingresso un numero di n cifre che di questo calcoli:

- tutte le possibili permutazioni.
- i numeri così ottenuti devono poi essere visualizzati contemporaneamente sullo schermo.

Scrivere il programma in modo che funzioni fino a n=5.

### 33. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che accetti in ingresso una sequenza di 16 cifre esadecimali. A partire dai caratteri di ingresso:

- formare la sequenza di 8 byte generata dal codice inserito.
- stampare la sequenza di 0 e di 1 così ottenuta.
- data la stringa prima ottenuta visualizzarne una codifica run length.

Per codifica run length si intende:

- Il primo bit della sequenza entra direttamente nella codifica.
- Segue la lunghezza espressa in decimale del numero di volte che il primo bit è presente nella stringa.
- Segue la lunghezza espressa in decimale del numero di volte che il complemento compare nella stringa e così via.

Esempio:

Sequenza ASCII: FFFF.0000.1010.0000

Sequenza di bit: 1111111111111111.0000000000000000.000100000010000.0000000000000000

Codifica: 1 16 19 1 7 1 20

### 34. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso un numero intero positivo K ( $K < 10000$ ).
- Calcoli la scomposizione in fattori primi del numero e la stampi.

### 35. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso un numero intero positivo K ( $K < 50000$ ).
- Accetti un secondo numero positivo n ( $n < 200$ ),
- Verifichi se K e' divisibile per n:

Se il numero e' divisibile si stampi il risultato.

Se il numero non e' divisibile si stampi il primo numero superiore e il primo numero inferiore (rispetto a n) che divide K in modo esatto.

### 36. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso una frase composta da 10 parole (max 10 caratteri)
- Si considerino le parole formate utilizzando tutti i caratteri alfanumerici (A ... Z, 0 ... 9).
- Stampi le parole inserite in ordine alfabetico crescente. I numeri seguono le lettere nell'ordinamento.

### 37. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso 10 numeri nel seguente formato nnnnEkk con il seguente significato:  
 $nnnn \times 10^{kk}$ .
- Stampi i numeri inseriti in ordine crescente.

### 38. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso un numero di 4 cifre.
- Stampi tutti i numeri (di 4 cifre) diversi tra loro che è possibile formare con le cifre del primo numero. (I numeri vanno stampati in modo che compaiano in una sola schermata).
- Stampi i numeri in ordine crescente (questo punto può essere svolto in una fase successiva).

### 39. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che:

- Accetti in ingresso un numero di 5 cifre.
- Stampi tutti i numeri (di 2 cifre) diversi tra loro che è possibile formare con le cifre del primo numero. (I numeri vanno stampati in modo che compaiano in una sola schermata).

- Stampi i numeri in ordine crescente (questo punto può essere svolto in una fase successiva).

#### 40. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 70 caratteri;  
La stringa può essere formata solo dai seguenti caratteri: 0,1,2,...,9 e A,B,C,...,Z (numeri e lettere); non possono esserci due caratteri 0,1,2,...,9 adiacenti; la stringa non può terminare con un numero.
- Conti le ricorrenze dei caratteri A,B,C,...,Z nella stringa; se un carattere viene preceduto da un numero n, si consideri come se tale carattere fosse ripetuto n volte consecutive.
- Stampi il carattere (o i caratteri) ripetuto(i) più frequentemente.

Introdurre il controllo della corretta sintassi e generare errore se le specifiche del primo punto non sono soddisfatte.

Esempio:

Sequenza in ingresso: ab4abbbm

Output: a

#### 41. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 70 caratteri;  
La stringa può essere formata solo dai seguenti caratteri: A,B,S,D, 2,...,9; non possono esserci due numeri adiacenti; la stringa non può terminare con un numero.
- Cancelli lo schermo, e stampi un cursore (carattere a scelta) al centro dello schermo stesso;
- Scandisca la stringa e sposti il cursore senza cancellare la posizione precedente secondo le seguenti regole:
  - Il carattere 'A' sposta in Alto il cursore di una posizione;
  - Il carattere 'B' sposta in Basso il cursore di una posizione;
  - Il carattere 'S' sposta a Sinistra il cursore di una posizione;
  - Il carattere 'D' sposta a Destra il cursore di una posizione;
  - Un numero agisce da prefisso di ripetizione per il carattere seguente.

In un primo tempo assumere che la stringa in ingresso soddisfi a tutte richieste e che il cursore non esca mai dallo schermo. Successivamente introdurre il controllo della corretta sintassi e generare errore se le specifiche non sono soddisfatte.

Esempio:

Sequenza in ingresso: as4a2sbbbs

Il cursore si deve muovere: asaaaassbbbs e deve generare un disegno del tipo:

```

      X X X
      X  X
      X  X
     X X  X
  
```

XX  
O

## 42. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 80 caratteri; La stringa può essere formata solo dai seguenti caratteri: 0, 1, 2,..., 9, +, -, = e deve rispettare le seguenti specifiche:
  - non possono comparire più di 3 caratteri numerici adiacenti;
  - i simboli +, -, = non possono essere adiacenti;
  - il simbolo = deve comparire una sola volta e al termine della stringa;Tale stringa forma così una espressione algebrica da valutare.
- Stampi il risultato della espressione algebrica, assumendo che tutti i risultati parziali possano essere memorizzati in una parola da 16 bit.

Verificare la sintassi della stringa e generare errore se le specifiche non sono soddisfatte.

Esempio:

Stringa in ingresso: 4+93-3-0+08=

Uscita: 4+93-3-0+08= 102

## 43. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 70 caratteri. La stringa può essere formata solo dai seguenti caratteri: numeri, lettere da 'A' ad 'F', e i seguenti simboli: +, \*, !. La stringa deve terminare con il segno '='.
- Calcoli il risultato della espressione logica assumendo che il simbolo '+' indichi l'operazione di OR, il simbolo '\*' quella di AND, e il simbolo '!' quella di EX-OR.
- stampi la stringa inserita e il risultato.

Assumere che la stringa in ingresso abbia, ad esempio, il seguente formato:

XXXXOXXXX.....XXXXOXXXX=

dove X indica una cifra esadecimale (range '0'-'F') e O indica una operazione logica (range: '+', '\*', '!').

La priorità nello svolgimento delle operazioni è da sinistra a destra.

Esempio:

Sequenza in ingresso: 1234+5678!9ABC=

Uscita: 1234+5678+9ABC=DEFC

## 44. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 60 cifre decimali;
- Accetti in ingresso una seconda stringa di 4 cifre decimali;
- Calcoli la funzione di correlazione tra le due stringhe, definita come:

per ogni posizione della seconda stringa rispetto alla prima calcolare il numero di caratteri coincidenti.

- Stampi la stringa ottenuta (la cui lunghezza deve essere uguale a quella della stringa in ingresso).

Assumere che la seconda stringa abbia lunghezza variabile (tra 2 e 9 caratteri).

Esempio:

Prima stringa in ingresso: 123412431434

Seconda stringa in ingresso: 1234

Uscita: 400121103000

#### 45. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso due date (giorno/mese/anno) nel formato: gg/mm/aa, dove:
  - gg varia da 01 a 30
  - mm varia da 01 a 12
  - aa varia da 1950 a 2050
- Calcoli e stampi il numero di giorni intercorsi tra le due date.

Esempio:

Prima data: 01 /12/1994

Seconda data: 13 /10 /1995

Giorni trascorsi: ????

#### 46. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso due orari (ore/minuti/secondi) nel formato: hh:mm:ss, dove
  - l'intervallo di variabilità per i minuti e i secondi :
    - mm varia da 00 a 59
    - ss varia da 00 a 59
- Calcoli e stampi il numero di secondi intercorsi tra i due orari.

Considerare anche il caso in cui il primo orario sia posteriore al secondo (cioè si riferisca al giorno precedente)..

Esempio:

Primo orario: 10:20:30

Secondo orario: 13:25:05

Secondi trascorsi: 11075

#### 47. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che continui ciclicamente a stampare i numeri da 0 a 100 mentre contemporaneamente è possibile premere opportune sequenze di tasti.

Considerare le possibili sequenze di tasti:

- "STOP" che termina il programma.
- "PAUSA" che ferma il ciclo dopo avere terminato di stampare il numero corrente.
- "VAI" che riprende il conteggio.

#### 48. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che, introducendo un giorno e un mese e un anno, determini il corrispondente giorno della settimana (si considerino gli anni dal 1950 al 2050).

Utilizzare un formato a piacere per l'introduzione dei dati e per la visualizzazione del risultato.

Esempio:

Giorno: 6

Mese: 5

Anno 2003

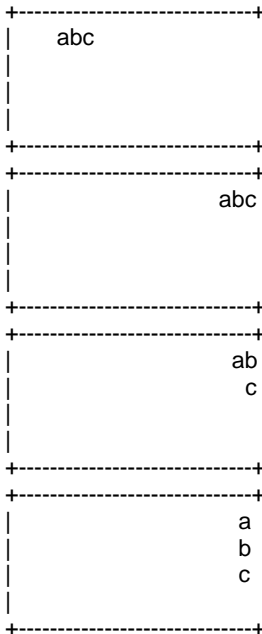
Il giorno della settimana è: Mercoledì

#### 49. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 20 caratteri e che continui ciclicamente a stamparla, facendola ruotare sullo schermo (matrice 80 \_ 25).

Più precisamente la stringa deve essere stampata sullo schermo prima alla posizione (0, 0), poi alla (0, 1), poi alla (0, 2) fintanto che non si raggiunge la posizione (0, 80 { <lunghezza della stringa>}); nella successiva stampa della stringa tutti i caratteri devono essere stampati consecutivamente, eccetto l'ultimo, che deve essere stampato sotto il penultimo.

```
+-----+
| abc   |
|       |
|       |
+-----+
```



Consentire al programma di eseguire più di un ciclo: al termine di ogni ciclo la stringa verrà raddoppiata fino ad un massimo di 30 caratteri.

Il programma termina con la pressione del tasto `SPAZIO' anche se la stringa non ha terminato il ciclo.

## 50. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 50 caratteri;
- Accetti in ingresso una seconda stringa numerica formata dallo stesso numero di caratteri della precedente.
- Cancelli lo schermo e stampi la prima stringa in alto a sinistra (posizione 0,0).

Il programma deve poi spostare ogni lettera della stringa appena stampata in basso di un numero di posizioni dato dal numero della seconda stringa associato alla lettera della prima stringa.

Esempio:

Prima stringa:       ABCDE

Seconda stringa:   12143

Output:             ABCDE  
                      A C  
                      B  
                          E  
                          D

Successivamente il programma deve entrare in un ciclo in cui le lettere della prima stringa rimbalzano tra le due posizioni indicate precedentemente.

## 51. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di 10 caratteri formata solo dai caratteri 'A', 'B', 'S', 'D';
- Stampi la stringa al centro dello schermo;
- Entri in un ciclo in cui ad ogni iterazione vengono mossi tutti i caratteri della stringa nel seguente modo:
  - i caratteri 'A' vengono spostati di una posizione verso l'Alto;
  - i caratteri 'B' vengono spostati di una posizione verso il Basso;
  - i caratteri 'S' vengono spostati di una posizione verso Sinistra;
  - i caratteri 'D' vengono spostati di una posizione verso Destra;
- Qualora la posizione in cui un carattere deve essere mosso risulti occupata, il carattere rimarrà fermo.
- Quando un carattere raggiunge il bordo dello schermo, si ferma.
- Il ciclo termina quando tutti i caratteri sono fermi.

Esempio:

```
.....  
.....  
.....ABSDAAS.....  
.....  
.....  
.....  
.....A..AA.....  
.....S..DS.....  
.....B.....  
  
.....  
.....A..AA.....  
.....  
.....S..DS.....  
.....  
.....B.....  
  
.....A..AA.....  
.....  
.....S..DS.....  
.....  
.....B.....  
  
.....A..AA.....  
.....  
.....S..DS.....  
.....  
.....B.....
```

## 52. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

Calcolatori elettronici A - 20/05/2009



- Accetti in ingresso una stringa composta da almeno 10 caratteri e massimo 70; la stringa deve contenere 2 lettere maiuscole mentre le rimanenti devono essere minuscole.
- Entri in un ciclo la cui funzione è di stampare la stringa inserita con le seguenti modalità:
  - le due lettere maiuscole si spostano all'interno della stringa di una posizione;
  - nel caso la posizione successiva sia già occupata dall'altra lettera maiuscola o se si è arrivati al termine della stringa, la lettera modificherà la propria direzione.
- Ad ogni iterazione il programma deve acquisire un carattere da tastiera: nel caso in cui il tasto sia "SPAZIO", il programma termina, altrimenti viene rieseguito il ciclo.

Considerare il controllo sulla correttezza del formato della stringa inserita.

Esempio:

```
aBcdefGhijkl
abCdefgHijkl
abcDefghijkl
abcdEfghiJkl
abcdeFghijKl
abcdefGhijkL
abcdefgHijKL
abcdefghIjkl
abcdefgHijKL
abcdefGhijkL
abcdeFghijKL
abcdEfghiJkl
abcDefghijkl
abCdefgHijkl
```

### 53. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly formato da un ciclo principale in cui un carattere viene mosso di una posizione per ogni iterazione a partire dall'angolo alto a sinistra dello schermo. Il carattere si sposterà verso destra fino al bordo destro dove scenderà di una posizione e continuerà a spostarsi verso sinistra.

Una volta arrivato al bordo sinistro, scenderà di una posizione e riprenderà a spostarsi verso destra.

Quando il carattere raggiunge l'ultima posizione dello schermo (angolo in basso a destra), riprenderà dalla posizione iniziale.

Inserire all'interno del ciclo due procedure che permettano all'utente di variare la velocità di movimento del carattere attraverso la pressione di due diversi tasti.

Attenzione: il carattere DEVE muoversi di una posizione ad ogni iterazione, indipendentemente dalla pressione dei tasti sopra menzionati!

Successivamente introdurre altre due procedure che permettano all'utente di:

- riportare il carattere alla posizione iniziale.
- fermare il movimento del carattere fino alla pressione di un altro tasto;

## 54. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- accetti in ingresso una stringa di massimo 80 caratteri;
- pulisca lo schermo;
- stampi la stringa inserita lettera per lettera a partire dal centro dello schermo spostandosi ad ogni carattere di una posizione verso destra, con le seguenti eccezioni:
  - ogni volta che viene stampata la lettera 'b', la stampa della stringa prosegue verso il basso dello schermo;
  - ogni volta che viene stampata la lettera 'a', la stampa della stringa prosegue verso l'alto dello schermo;
  - ogni volta che viene stampata la lettera 'd', la stampa della stringa prosegue verso la destra dello schermo;
  - ogni volta che viene stampata la lettera 's', la stampa della stringa prosegue verso la sinistra dello schermo.
- introdurre il controllo che la stampa non esca dallo spazio dello schermo; in tal caso segnalare errore.
- si stampi un asterisco nel caso in cui la posizione corrente sia già occupata da un altro carattere.

Inserire una stringa: hujaikosusjhurbfuhedplpp

```
bruhjsus
f      o
u      k
h      i
e huj
dplpp
```

## 55. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- accetti in ingresso una stringa composta da massimo 70 caratteri; la stringa può contenere solo lettere maiuscole;
- entri in un ciclo in cui ad ogni iterazione vengano stampate due nuove stringhe: una ottenuta dalla precedente eliminando le lettere (anche multiple) con codice ASCII inferiore, mentre la seconda deve contenere le lettere eliminate dalla prima.
- il ciclo termina quando la stringa risulta vuota.

Esempio:

```
Inserire una stringa:  FFAFCAAMFFFEEEEACF
iterazione 1:         FFFCMFFFEEECF      AAAA
iterazione 2:         FFFMFFFEEEF      AAAACC
```

iterazione 3:	FFFMFFFF	AAAACCEEE
iterazione 4:	M	AAAACCEEEFFFFFFF
iterazione 5:		AAAACCEEEFFFFFFFM

## 56. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- accetti in ingresso una stringa di massimo 70 caratteri composta da lettere maiuscole, minuscole e spazi; lo spazio, che può anche comparire più volte consecutivamente, è carattere separatore tra parole diverse;
- accetti in ingresso una lettera maiuscola;
- stampi tutte le parole della prima stringa che contengono la lettera di controllo specificata dalla seconda immissione su righe diverse consecutivamente, in modo da allineare sulla stessa colonna (centrata nello schermo) la lettera di controllo.

Nel caso si avessero più ricorrenze della lettera di controllo in una stessa parola, tale parole viene considerata e stampata più volte.

Successivamente eliminare l'inserimento della lettera di controllo e creare 5 colonne, una per ogni vocale.

Esempio:

Inserire una stringa: Compito di Calcolatori Elettronici

Inserire una lettera: T

```

      *
      Compito
      Calcolatori
      Elettronici
      Elettronici

      a          e          i          o
      Calcolatori Elettronici Compito di Compito
      Calcolatori Elettronici Calcolatori Calcolatori
      Elettronici Calcolatori Calcolatori
      Elettronici Elettronici
  
```

## 57. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- accetti in ingresso 10 numeri  $n_i$ ,  $i = 1 \dots 10$ , compresi tra 0 e 1000;
- rappresenti i numeri inseriti sotto forma di istogramma (a tal fine si scelga un modo per la loro approssimazione).

Considerare due situazioni diverse:

- assumere che il range di variabilità dell'istogramma sia tra 0 e 1000;



lettera digitata; la nuova stringa viene sovrascritta alla precedente (alle stesse coordinate della precedente), eliminando gli eventuali caratteri rimasti dalla stringa precedente.

Successivamente modificare il programma in modo che la pressione di tasti corrispondenti a lettere minuscole abbiano l'effetto di eliminare le corrispondenti ricorrenze nella stringa iniziale (come sopra), mentre la pressione di tasti corrispondenti a lettere maiuscole abbiano l'effetto opposto, ovvero di reinserire le lettere eliminate in precedenza.

Esempio:

Inserire una stringa: abcAAbCCaBBc

abcAAbCCaBBcdE

(dopo la pressione di 'a')

bcbCCBBcdE

(dopo la pressione di 'b')

cCCcdE

(dopo la pressione di 'c')

dE

(dopo la pressione di 'B')

bbBBdE

(dopo la pressione di 'A')

abAAbBBdE

(dopo la pressione di 'C')

abcAAbCCaBBcdE

## 60. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 80 caratteri formata da insiemi di parole di massimo 6 caratteri intervallate da uno spazio. Le parole siano formate da lettere maiuscole o minuscole.

Si consideri il testo formato dalla ripetizione per 10 volte della stringa inserita.

Il programma deve stampare a schermo tale testo, giustificandolo a destra e a sinistra, ovvero inserendo opportunamente degli spazi per fare in modo che le prime e le ultime lettere di ogni riga siano incolonnate.

Nella stampa finale ogni riga deve contenere 80 caratteri e non vi possono essere più di due spazi adiacenti.

Successivamente modificare il programma in modo che venga richiesta all'utente la dimensione della riga di testo in uscita.

Inserire una stringa: Nel mezzo del cammin di nostra vita

Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita Nel mezzo del cammin di nostra vita

## 61. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 80 caratteri formata da insiemi di parole intervallate da uno o più spazi. Le parole siano formate da lettere maiuscole o minuscole.

Il programma deve creare due stringhe diverse formate dalla concatenazione delle parole composte da un numero pari di lettere e da quelle formate da un numero dispari di lettere. Le parole devono essere intervallate da un solo spazio. Le due stringhe devono poi essere stampate.

Successivamente ricostruire una sola stringa in cui siano intervallate successivamente una parola dalla prima stringa e una dalla seconda. Dapprima considerare le due stringhe formate da un ugual numero di parole. In seguito, nel caso una stringa sia formata da un numero di parole diverso dall'altra, ripetere al termine della stringa più corta l'ultima parola per un numero di volte tale per cui entrambe le stringhe consistano dello stesso numero di parole.

Inserire una stringa:

abc ABCDE abcdefgh AB abc xyz XYX

parole di lunghezza pari: abcdefgh AB

parole di lunghezza dispari: abc ABCDE abc xyz XYX

stringa finale: abcdefgh abc AB ABCDE AB abc AB xyz AB XYX

## 62. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una sequenza di numeri ad una cifra (caratteri). La sequenza deve contenere minimo 2 numeri e massimo 20 numeri. Il programma deve accettare i primi due numeri e successivamente deve comportarsi nel seguente modo:

- se la somma dei due ultimi numeri è pari, deve accettare il numero solo se esso è maggiore o uguale al precedente;
- se la somma dei due ultimi numeri è dispari, deve accettare il numero solo se esso è minore o uguale al precedente;

Il programma deve terminare automaticamente una volta inserito il ventesimo numero.

In caso di errore il programma deve suggerire una possibile soluzione.

Successivamente modificare il programma in modo che se è possibile inserire solo un carattere esso venga immesso automaticamente.

Inserire una sequenza di numeri ad una cifra:

1 2 1 0 0 8 9 7 8 5 3 9 9 9 9

## 63. CALCOLATORI ELETTRONICI

Scrivere un programma che permetta di inserire una stringa di esattamente 25 caratteri numerici. Il programma deve trattare questo insieme di caratteri come gli elementi di una matrice 5 x 5, stampare questa matrice, e far evolvere gli elementi come descritto nel seguito:

Ogni elemento  $a_t(i,j)$  (riga  $i$ , colonna  $j$ , al tempo  $t$ ) deve essere calcolato come media tra i primi vicini:

$$a_{t+1}(i,j) = \frac{a_t(i+1,j) + a_t(i-1,j) + a_t(i,j+1) + a_t(i,j-1)}{4}$$

4

(1)

con  $i = 1 \dots 3$  e  $j = 1 \dots 3$ .

Si consideri il range di variabilità degli elementi tra 0 e 9. Gli elementi delle colonne 0 e 4 e delle righe 0 e 4 non cambiano.

Il programma deve continuare ad iterare fintanto che non interviene più nessuna modifica nella matrice.

Successivamente modificare il programma in modo che varino anche gli elementi delle colonne 0 e 4 e delle righe 0 e 4, considerando che gli elementi delle colonne esterne -1 e 5 e delle righe esterne -1 e 5 siano nulli.

Inserire una stringa: 1234567890123456789012345

Matrice:

1 2 3 4 5

6 7 8 9 0

1 2 3 4 5

6 7 8 9 0

1 2 3 4 5

Iterazione 1:

1 2 3 4 5

6 4 5 4 0

1 4 5 6 5

6 4 5 4 0

1 2 3 4 5

## 64. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 80 caratteri formata da insiemi di parole intervallate da uno o più spazi. Le parole siano formate da lettere maiuscole o minuscole.

Il programma deve ordinare le parole in modo decrescente rispetto al:

- numero di adiacenze tra vocali e consonanti in una stessa parola;
- numero di adiacenze tra numeri e lettere in una stessa parola;
- numero di consonanti in una stessa parola.

Inserire una stringa:

QUESTO TEST23 9SERVE9 COME ESEMPIO

Ordinamento rispetto al numero di adiacenze tra vocali e consonanti in una stessa parola:

ESEMPIO

QUESTO

9SERVE9

COME

TEST23

Ordinamento rispetto al numero di adiacenze tra numeri e lettere in una stessa parola:

9SERVE9

TEST23

QUESTO

COME

## ESEMPIO

Ordinamento rispetto al numero di consonanti in una stessa parola:

QUESTO  
TEST23  
9SERVE9  
ESEMPIO  
COME

## 65. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che:

- Accetti in ingresso una stringa di massimo 80 caratteri;
- Accetti in ingresso una seconda stringa di massimo 8 cifre numeriche crescenti;
- Per ogni cifra della seconda stringa stampi un numero di caratteri della prima stringa pari al valore della cifra;

La prima stampa deve procedere verso destra a partire dal centro dello schermo, ogni stampa successiva deve essere effettuata in una direzione ruotata di 90° in senso orario rispetto alla precedente.

In un primo tempo supporre che i dati soddisfino le specifiche, successivamente effettuare l'ordinamento della seconda stringa, se necessario, e i controlli opportuni.

Esempio:

**Prima stringa: nelmezzodelcammindinostravitamiritrovaiinunaselvaoscura**

**Seconda stringa: 24578**

```
dinostrav
n
i
m ne
m l
a m
c e
ledozz
```

## 66. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che consenta di tracciare un triangolo sullo schermo in modalità testo mediante la seguente procedura:

- spostare un cursore usando le frecce per definire i vertici del triangolo;
- inserire nella posizione del cursore (dopo aver utilizzato un carattere a scelta per entrare in modalità di inserimento) i vertici del triangolo;
- con un ulteriore comando costruire il triangolo definito dai tre vertici;
- un comando successivo genera lo riempimento dell'area del triangolo con un carattere a scelta.



## 67. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 80 caratteri formata da parole fatte di lettere maiuscole o minuscole e intervallate da spazi.

Il programma deve:

- stampare il numero di consonanti contenute in ogni parola e ordinare le parole rispetto al numero decrescente di consonanti;
- ristampare le parole scrivendo prima tutte le consonanti e poi tutte le vocali nell'ordine in cui compaiono nella parola (separare il gruppo delle consonanti dal gruppo delle vocali con un asterisco);

Esempio:

**Inserire una stringa:**

**Corso di Laurea in Ingegneria Elettronica Informatica delle Telecomunicazioni**

**3 1      2 1          5          6          6 3                  8**

**Telecomunicazioni Elettronica Informatica Ingegneria Corso delle Laurea di in**

**Tlcmnczn\*eeouiaioi ltrnc\*Eeoia nfrmtc\*loaia nggnr\*leeia Crs\*oo dll\*ee Lr\*auca d\*i n\*i**

## 68. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che accetti in ingresso una stringa di massimo 255 caratteri contenente più parole formate da lettere minuscole o maiuscole separate tra di loro da uno o più spazi o da eventuale punteggiatura, apostrofi etc.

Il programma deve stampare a schermo le parole contenute nel testo fornito e ordinarle per lunghezza.

A parità di lunghezza le parole devono essere ordinate alfabeticamente.

Esempio:

**Inserire una stringa: lasciate ogni speranza voi che entrate**

**che  
voi  
ogni  
entrate  
lasciate  
speranza**

## 69. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che acquisisca in ingresso una stringa di massimo 80 caratteri costituita da più *parole* separate tra di loro da spazi.

Si supponga che le parole siano costituite solamente da caratteri (maiuscoli o minuscoli) o cifre.

Il programma deve calcolare e stampare il numero di occorrenze nella stringa di ciascun carattere o cifra. La stampa deve avvenire seguendo l'ordine alfabetico delle sole lettere o numeri presenti. I numeri devono precedere le lettere.

## 70. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che iterativamente accetti in ingresso parole al massimo di 40 caratteri formate esclusivamente da lettere maiuscole o minuscole.

Il programma deve stampare a schermo le parole:

- così come appaiono nella stringa in ingresso se il numero di lettere maiuscole è maggiore o uguale al numero di lettere minuscole,
- invertendo l'ordine delle lettere nel caso in cui il numero di lettere minuscole sia maggiore del numero di lettere maiuscole.
- Le lettere in ordine alfabetico indipendentemente dal fatto che siano maiuscole o minuscole.

Il programma termini qualora venga premuto il tasto ``%".

## 71. CALCOLATORI ELETTRONICI

Uno dei problemi classici dell'informatica è la risoluzione di semplici espressioni come, ad esempio:  $((10-5)+((8-2)-(3+4)))$ .

Infatti, la presenza di parentesi complica la risoluzione costringendo al calcolo e alla relativa memorizzazione di risultati parziali.

La risoluzione di espressioni di questo tipo risulta più semplice se le si scrive in notazione polacca inversa che non richiede l'utilizzo di parentesi. Nel caso visto in precedenza:

$10\ 5\ -\ 8\ 2\ -\ 3\ 4\ +\ -\ +\ .$

### **Esercizio:**

scrivere un programma assembly che prende in ingresso espressioni e le traduca in notazione polacca inversa.

Si ipotizzi che le espressioni non contengano numeri negativi e che le sole operazioni siano la somma (+) e la sottrazione (-).

Successivamente implementare anche il calcolo del risultato ipotizzando che i numeri e i risultati (anche parziali) abbiano al più 2 cifre.

### **Suggerimento:**

La trasformazione di un'espressione in notazione polacca inversa si può facilmente implementare mediante l'utilizzo dello stack. Occorre scandire l'espressione da sinistra a destra e stampare (o memorizzare) i numeri che si incontrano. Quando si incontra una operazione questa va posta sullo stack. Viceversa, nel caso si arrivi ad una parentesi chiusa occorre prelevare l'operazione che si trova nello stack e stamparla.

Analogamente, anche il calcolo del risultato a partire dalla notazione polacca inversa si può implementare sfruttando lo stack. Anche in questo caso occorre scandire l'espressione da sinistra a destra e porre eventuali numeri che si incontrano sullo stack. Quando si arriva ad una operazione occorre prelevare i due numeri in cima allo stack, effettuare il calcolo e riporre il risultato sullo stack.

Al termine il risultato si troverà in cima allo stack.

### **Esempio:**

**introdurre un espressione: ((10-5)+((8-2)-(3+4)))**  
**notazione inversa: 10 5 - 8 2 - 3 4 + - +**  
**risultato: 4**

## 72. CALCOLATORI ELETTRONICI

Scrivere un programma assembly che acquisisca in ingresso le coordinate di due punti e disegni sul video il rettangolo individuato dai due punti.

Si scelga di lavorare in una modalità testuale e si disegni il rettangolo con un carattere a scelta.

Si supponga che le coordinate fornite in ingresso siano corrette rispetto alla risoluzione scelta.

Successivamente si modifichi il programma in maniera che disegni un carattere (o una pallina) all'interno del rettangolo che si muova in diagonale e rimbalzi contro le pareti.

**Esempio:**

**introdurre ascissa angolo alto/sinistra: 2**

**introdurre ordinata angolo alto/sinistra: 3**

**introdurre ascissa angolo basso/destra: 15**

**introdurre ordinata angolo basso/destra: 20**

## 73. CALCOLATORI ELETTRONICI

Un polinomio di grado generico può essere rappresentato come una sequenza di coefficienti, ad esempio può essere rappresentato come:

8 -2 0 7 0 5

Scrivere un programma assembly che acquisisca in ingresso due sequenze di coefficienti relative a due polinomi e ne calcoli la somma:

Si supponga che i coefficienti siano interi anche negativi compresi tra 9 e -9.

Si calcoli il valore del polinomio per  $x=n$ .

**Esempio:**

**Introdurre primo polinomio: 2 -3 0 4**

**Introdurre secondo polinomio: 0 8 4 5 6**

**Primo polinomio:  $4x^3-3x+2$**

**Secondo polinomio:  $6x^4+5x^3+4x^2+8x$**

**Somma dei polinomi:  $6x^4+9x^3+4x^2+5x+2$**

**Introdurre valore per x: 8**

**Risultato per x=8: 2590**

## 74. CALCOLATORI ELETTRONICI

Il codice fiscale utilizzato in Italia è formato da 16 caratteri alfanumerici che codificano alcune informazioni sulla persona a cui fa riferimento il codice. Nell'ordine si hanno:

- 3 lettere derivanti dal cognome (tipicamente consonanti),
- 3 lettere derivanti dal nome (tipicamente consonanti),
- le ultime due cifre dell'anno di nascita,
- una lettera che codifica il mese (A=gennaio; B=febbraio; C=marzo; D=aprile; E=maggio; H=giugno; L=luglio; M=agosto; P=settembre; R=ottobre; S=novembre; T=dicembre),
- le due cifre del giorno di nascita a cui viene sommato 40 per le donne;
- 4 caratteri alfanumerici (nell'ordine una lettera e tre numeri) che codificano il comune di nascita,
- una lettera finale di verifica del codice.

Ad esempio Teodora Arguti nata a Parma il 7 ottobre 1982 avrebbe come codice: RGTDR82R47G337Q

### **Esercizio:**

Scrivere un programma assembly che effettui un controllo (molto semplificato) di congruenza del codice fiscale.

Il programma dovrà prendere in ingresso un codice fiscale ed effettuare i seguenti controlli:

- Verificare che il codice sia formato da 16 caratteri alfanumerici (solo lettere maiuscole e numeri);
- Verificare che i primi sei caratteri siano consonanti;
- Verificare che i successivi due caratteri siano numeri e stampare il relativo anno di nascita;
- Verificare che la lettera che codifica il mese di nascita sia congruente con le indicazioni fornite e stampare il relativo mese;
- Verificare che i successivi due caratteri siano numeri e stampare giorno di nascita e sesso indicati, controllare che il giorno di nascita sia congruente;
- Verificare che i successivi 4 caratteri siano rispettivamente una lettera e tre cifre;
- Verificare che l'ultimo carattere sia una lettera.

Esempio:

Introdurre codice fiscale: RGTDR82R87G337Q

Verifica lunghezza: corretto

Verifica cognome: corretto

Verifica nome: corretto

Verifica anno: 1982

Verifica mese: ottobre

Verifica giorno e sesso: ERRORE!

## 75. CALCOLATORI ELETTRONICI

Scrivere un programma in assembler che effettui il prodotto di due matrici  $A$  e  $B$  e dia come risultato la matrice  $C$ .

Il programma deve:

- Accettare da tastiera le due matrici (gli elementi delle matrici  $A$  e  $B$ , sono numeri positivi inferiori a 10.)
- Presentare su video la matrice  $C$  risultato del prodotto.

Scrivere il programma nel caso di matrici di dimensione  $3 \times 3$ .

## 76. CALCOLATORI ELETTRONICI

Una delle più antiche (e deboli) tecniche di cifratura è la cifratura a sostituzione monoalfabetica nota anche come cifratura di Cesare. Questa tecnica si basa sulla sostituzione delle lettere/simboli di un messaggio mediante l'utilizzo di un *alfabeto* alternativo. Ad esempio, nella seguente frase, riportata sia in chiaro che cifrata, tutte le "a" sono state sostituite con "n", tutte le "s" con "f", etc.

Esempio di codifica:

Sul sito trovate il pacchetto RPM, i sorgenti TGZ, documentazione e manuale.

Fhy fvgb gebingr vy cnpurggb ECZ, v fbetravg GTM, qbphzragnmvbar r znahnyr.

### Esercizio:

Scrivere un programma in assembler che prenda in ingresso una stringa di massimo 80 caratteri e la cifri mediante la tecnica della cifratura monoalfabetica.

Inizialmente si inserisca l'alfabeto cifrante direttamente nel codice. Successivamente, si integri il programma in maniera da prendere in ingresso l'alfabeto cifrante (chiave).

### Esempio:

Inserire la frase da cifrare.

-> Tanto va la gatta al lardo che ci lascia lo zampino

Inserire la chiave per l'alfabeto "abcdefghijklmnopqrstuvwxyz":

-> nopqrstuvwxyzabcdefghijklmnop

Frase cifrata:

-> Gnagb in yn tnggn ny yneqb pur pv ynfavn yb mnzcvab

## 77. CALCOLATORI ELETTRONICI

Una funzione importante dei sistemi di elaborazione testi è la giustificazione del testo, cioè l'allineamento delle parole ai margini sinistro e destro della pagina. In generale questo avviene inserendo degli spazi tra le parole del paragrafo da giustificare, in maniera che l'ultima parola di ogni riga termini sul margine destro.

### Esercizio:

Scrivere un programma in assembler che prenda in ingresso una stringa di massimo 200 caratteri e la giustifichi ipotizzando una larghezza testo di 40 caratteri. Si distribuiscano gli spazi bianchi necessari nel modo più uniforme possibile, rispettando comunque la regole che non si mette mai uno spazio all'inizio della linea, e che non si mettono mai spazi davanti ai segni di punteggiatura.

**Esempio:**

Inserire la stringa da giustificare:

Cantami, o Diva, del Pelide Achille l'ira funesta che infiniti addusse lutti agli Achei, molte anzi tempo all'Orco generose travolse alme d'eroi,

Risultato:

Cantami, o Diva, del Pelide Achille  
l'ira funesta che infiniti addusse lutti  
agli Achei, molte anzi tempo all'Orco  
generose travolse alme d'eroi,

**78. CALCOLATORI ELETTRONICI**

Scrivere un programma che esegua la funzione di scrittura analoga a quella dei telefonini per la scrittura di messaggi. Le cifre da 0 a 9 associano a ciascun tasto le lettere dell'alfabeto come da tabella seguente:

1	Spazio
2	ABC
3	DEF
4	GHI
5	JKL
6	MNO
7	PQRS
8	TUV
9	WXYZ

Quando si vogliono scrivere gli SMS la pressione singola o multipla di un tasto numerico permette di selezionare la lettera desiderata.

Ad esempio, per ottenere la lettera R occorre premere 3 volte il tasto 7.

Simulare il tasto di fine con il tasto di spazio e la cancellazione con il tasto *canc.*

**79. CALCOLATORI ELETTRONICI**

Una delle più semplici tecniche di compressione dati (peraltro utilizzata nel formato grafico PCX) è quella basata sull'individuazione di ripetizioni nei dati stessi e nella loro sostituzione con sequenze più brevi. Ad esempio, la stringa ``AAAADBBBCCAA" può essere codificata mediante la stringa ``4AD3B2C2A" in cui i prefissi numerici hanno il significato di ripetizione.

**Esercizio:**

Scrivere un programma in assembly che prenda in ingresso una stringa di massimo 80 caratteri contenente solo lettere e che stampi a video la codifica di questa stringa ottenuta calcolando le ripetizioni di ciascun carattere e stampando ciascun carattere una sola volta preceduto dal numero delle ripetizioni.

Risolvere anche il problema inverso.

**Esempio:**

Inserire una stringa: AAADBBBCCAAA  
Versione codificata: 3AD3BCC3A

80. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che prenda in ingresso una stringa di massimo 80 caratteri costituita soli numeri [0,1,2,3 ...,9].

Il programma analizzi i dati in ingresso individuando la sequenza più lunga di numeri disposti in sequenza crescente e la stampi.

**Esempio:**

Inserire una stringa: 1391346283940132  
Risultato: 1346

81. CALCOLATORI ELETTRONICI

Scrivere un programma in Assembly che prenda in ingresso una stringa di massimo 80 caratteri costituita solo da lettere minuscole [a,z].

Il programma elabori i dati in ingresso individuando la sequenza più lunga di lettere disposte in sequenza crescente rispetto all'ordine alfabetico e la stampi.

**Esempio:**

Inserire una stringa: askahdffghjedefjoqtasdfgeqwr  
Risultato: adefjoqt

82. CALCOLATORI ELETTRONICI

Le tastiere dei telefoni cellulari, oltre a contenere le cifre da 0 a 9 associano a ciascun tasto le lettere dell'alfabeto come da tabella seguente:

1	Spazio
2	ABC
3	DEF
4	GHI
5	JKL
6	MNO
7	PQRS
8	TUV
9	WXYZ

Quando si vogliono scrivere gli SMS la pressione singola o multipla di un tasto numerico permette di selezionare la lettera desiderata.

Ad esempio, per ottenere la lettera R occorre premere 3 volte il tasto 7.

#### **Esercizio:**

Scrivere un programma in Assembly che prenda in ingresso una stringa di massimo 80 caratteri costituita esclusivamente da parole composte da cifre separate tra loro da un singolo spazio. Ciascuna parola è costituita da una o più ripetizioni della stessa cifra.

Il programma elabori i dati in ingresso interpretando le parole come pressioni singola o multipla di un tasto numerico di un cellulare e stampi le lettere corrispondenti.

il programma deve controllare la correttezza dei dati in ingresso.

#### **Esempio:**

Inserire una stringa: 6 444 1 7 444 2 222 33

Risultato: MI PIACE

#### **83. CALCOLATORI ELETTRONICI**

Scrivere un programma in Assembly che acquisisca in ingresso un numero compreso tra 2 e 9.

Visualizzare sul video in formato decimale e binario tutte le potenze del numero in ingresso fino al valore massimo contenibile in 64 bit.

#### **84. CALCOLATORI ELETTRONICI**

Scrivere un programma in Assembly che acquisisca in ingresso una stringa di massimo 80 caratteri. Il programma deve stampare la codifica *run lenght* della stringa inserita considerandola in formato



binario. Ovvero, considerando i byte della stringa in formato binario, il programma analizza la stringa bit a bit e stampa:

1. inizialmente il valore del primo bit
2. a seguire, il numero di volte consecutive che quel bit appare nella stringa
3. il numero di volte consecutive che appare il complemento del primo bit
4. a ripetere dal punto (2) fino alla fine della stringa

Ad esempio se viene inserita la stringa "Ciao" che in binario è 1000011110100111000011101111 il programma deve stampare:

1 1 4 4 1 1 2 3 4 3 1 4

Il programma deve effettuare anche la conversione inversa.