

# CALCOLATORI ELETTRONICI

## Prova scritta

22 ottobre 90

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

- Il numero stesso in codice binario.
- Il suo quadrato sempre in codice binario.

# CALCOLATORI ELETTRONICI

## Prova scritta

3 giugno 1991

Scrivere un programma in assembler che:

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

# CALCOLATORI ELETTRONICI

## Prova scritta

17 luglio 1991

Scrivere un programma in assembler che:

- Accetti da tastiera una parola di 16 bit nella sua rappresentazione esadecimale.
- Visualizzi sullo schermo la rappresentazione esadecimale delle 16 configurazioni che si ottengono facendo circolare il dato precedente in un registro di 16 posizioni richiuso su se stesso.

# CALCOLATORI ELETTRONICI

## Prova scritta

29 luglio 1991

Scrivere un programma in assembler che:

- Accetti da tastiera numeri positivi e negativi di 4 cifre piu' 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.

# CALCOLATORI ELETTRONICI

## Prova scritta

16 settembre 1991

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.
  - stampare la lista delle parole già inserite.
  - uscire dal programma.

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

# CALCOLATORI ELETTRONICI

## Prova scritta

21 ottobre 1991

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.
  - richiedere 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.

# CALCOLATORI ELETTRONICI

## Prova scritta

25 novembre 1991

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:

- Presentarsi visualizzando gli impegni nel formato seguente:

9.00 .....

10.00 .....

11.00 .....

.....

17.00 .....

- Permettere l'inserimento di nuovi impegni e al termine la visualizzazione degli stessi.
- Permettere la cancellazione di tutti gli impegni.
- Permettere di uscire dal programma.

L'impegno orario e' rappresentato da un stringa alfanumerica (max 50 caratteri).

# CALCOLATORI ELETTRONICI

## Prova scritta

10 febbraio 1992

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  $B$ , di dimensione  $2 \times 2$ , sono numeri positivi inferiori a 10.)
- Presentare su video la matrice  $C$  risultato del prodotto.

Scrivere successivamente lo stesso programma nel caso di dimensione  $3 \times 3$ .

# CALCOLATORI ELETTRONICI

## Prova scritta

18 febbraio 1992

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

$$x_{i+1} = |ax_i + b|_{mod256}$$

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

Il programma deve:

- Acquisire i valori di  $a$  e  $b$ ;
- Visualizzare la sequenza su 20 righe di 20 numeri con il seguente formato:

bbb1bb25b145bb30b100bbb0...

# CALCOLATORI ELETTRONICI

## Prova scritta

7 aprile 1992

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.

# CALCOLATORI ELETTRONICI

## Prova scritta

1 giugno 1992

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 xxxx,xxxx. La trasformazione è ovviamente approssimata.

Il programma deve stampare come output:

- La trasformazione in base 2 xxxx,xxxx.
- La ritrasformazione in base 10 esatta del numero in base 2 calcolato al punto precedente.

# CALCOLATORI ELETTRONICI

## Prova scritta

17 luglio 1992

Scrivere un programma in assembler che, dato un numero compreso fra 0 e 999, 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:  $150 = 2 \ 3 \ 5(2)$

# CALCOLATORI ELETTRONICI

## Prova scritta

17 settembre 1992

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:

1.  $2 \leq K \leq 20$ .
2. Il valore del numero rappresentato da  $N(\text{base } K)$  deve essere minore di 65536 (decimale) =  $2^{16}$ ; in caso contrario il programma deve segnalare errore.
3. I numeri da usare per  $K$  sono:

$$2 \rightarrow 2, \dots 9 \rightarrow 9, 10 \rightarrow A, \dots 20 \rightarrow M.$$

4. Gli stessi simboli devono essere usati per rappresentare i numeri.  
Esempio: 23B (base C) = 335 (base A).

E' facoltativo estendere la condizione 2 fino a  $2^{32}$

# CALCOLATORI ELETTRONICI

## Prova scritta

19 novembre 1992

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   |
| radice cubica:   | 11   |
|                  | 1600 |
|                  | 1331 |

# CALCOLATORI ELETTRONICI

## Prova scritta

8 febbraio 1993

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.

In una prima fase realizzare il programma indicato considerando solo numeri positivi.

## CALCOLATORI ELETTRONICI

### Prova scritta

6 aprile 1993/1

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 e' opportuno inserire un ritardo; considerare la posizione piu' opportuna nel loop per il ritardo stesso.

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

# CALCOLATORI ELETTRONICI

## Prova scritta

6 aprile 1993/2

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 e' opportuno inserire un ritardo; considerare la posizione piu' opportuna nel loop per il ritardo stesso.

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

## CALCOLATORI ELETTRONICI

### Prova scritta

1 giugno 1993/1

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).

## CALCOLATORI ELETTRONICI

### Prova scritta

21 luglio 1993/1

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.

# CALCOLATORI ELETTRONICI

## Prova scritta

15 settembre 1993

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.

# CALCOLATORI ELETTRONICI

## Prova scritta

5 ottobre 1993

Scrivere un programma in assembler che:

1. Accetti da tastiera una stringa contenente al massimo 80 caratteri.
2. Accetti da tastiera una stringa contenente al massimo 4 caratteri.
3. 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 determinino 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).

# CALCOLATORI ELETTRONICI

## Prova scritta

3 novembre 1993

Scrivere un programma in assembler che:

1. Accetti da tastiera una stringa contenente 8 caratteri ASCII.
2. Visualizzi tale stringa in formato esadecimale.
3. 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.
4. Stampi la stringa di bit risultante (72 bit = 8 caratteri da 9 bit) in formato esadecimale.

## CALCOLATORI ELETTRONICI

### Prova scritta

11 gennaio 1994

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.

# CALCOLATORI ELETTRONICI

## Prova scritta

7 febbraio 1994

Scrivere un programma in assembler che accetti in ingresso un numero di  $n$  cifre (con  $n$  fino ad un massimo di 10) 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  $n=5$ .

Al fine di raggiungere la sufficienza è accettabile la versione del programma per  $n=4$ .

# CALCOLATORI ELETTRONICI

## Prova scritta

28 febbraio 1994

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: 11111111111111.0000000000000000.000100000010000.0000000000000000

Codifica: 1 16 19 1 7 1 20

# CALCOLATORI ELETTRONICI

## Prova scritta

13 giugno 1994/1

Scrivere un programma in assembler che:

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

# CALCOLATORI ELETTRONICI

## Prova scritta

13 giugno 1994/2

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 e' divisibile stampa il risultato.
  - Se non e' divisibile stampa il primo numero superiore e il primo numero inferiore (rispetto a  $n$ ) che divide  $K$  in modo esatto. (Limitarsi ai numeri compresi fra 1 e 200)

# CALCOLATORI ELETTRONICI

## Prova scritta

18 luglio 1994/1

Scrivere un programma in assembler che:

- Accetti in ingresso 10 parole (max 10 caratteri A...Z).
- Stampi le parole inserite in ordine alfabetico crescente.

In una seconda fase si considerino le parole formate utilizzando tutti i caratteri alfanumerici (A ... Z, 0 ... 9). I numeri seguono le lettere nell'ordinamento.

# CALCOLATORI ELETTRONICI

## Prova scritta

18 luglio 1994/2

Scrivere un programma in assembler che:

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

In una seconda fase si consideri la possibilità che il numero possa assumere i seguenti formati:  $n.nnnEkk$  oppure  $nn.nnEkk$  oppure  $nnn.nEkk$ .

# CALCOLATORI ELETTRONICI

## Prova scritta

5 settembre 1994/1

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).

# CALCOLATORI ELETTRONICI

## Prova scritta

5 settembre 1994/2

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).

# CALCOLATORI ELETTRONICI

## Prova scritta

28 Ottobre 1994

Scrivere un programma in Assembly che:

1. 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.

2. 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.
3. Stampi il carattere (o i caratteri) ripetuto(i) più frequentemente.

In un primo tempo assumere che la stringa in ingresso soddisfi alle richieste del punto

1. Successivamente introdurre il controllo della corretta sintassi e generare errore se le specifiche del punto 1. non sono soddisfatte.

Al termine, ripetere il ciclo con una seconda stringa, ma evidenziare il carattere (o i caratteri) che hanno variato maggiormente la loro frequenza nelle due stringhe.

Esempio:

Sequenza in ingresso: ab4abbm

Output: a

# CALCOLATORI ELETTRONICI

## Prova scritta

23 Gennaio 1995

Scrivere un programma in Assembly che:

1. 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.

2. Cancelli lo schermo, e stampi un cursore (carattere a scelta) al centro dello schermo stesso;
3. 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:

```
XXX
 X X
 X X
XX X
 XX
  0
```

# CALCOLATORI ELETTRONICI

## Prova scritta

12 Aprile 1995

Scrivere un programma in Assembly che:

1. 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 numeri 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.

2. cancelli lo schermo, e stampi la stringa inserita;
3. stampi il risultato della espressione algebrica, assumendo che tutti i risultati parziali possano essere memorizzati in una parola da 16 bit.

In un primo tempo assumere che la stringa in ingresso soddisfi le specifiche sopra elencate. Successivamente introdurre il controllo della corretta sintassi 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

# CALCOLATORI ELETTRONICI

## Prova scritta

16 Giugno 1995

Scrivere un programma in Assembly che:

1. 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 '='.

2. 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 EXOR.
3. stampi la stringa inserita e il risultato.

In un primo tempo assumere che la stringa in ingresso abbia il seguente formato:

XXXXOXXXXOXXXX=

dove X indica una cifra esadecimale (range '0'-'F') e O indica una operazione logica (range: '+', '\*', '^'). Successivamente considerare la stringa formata sempre dalla concatenazione di due operazioni logiche, ma i cui operandi non siano limitati a 4 cifre.

Esempio:

Sequenza in ingresso: 1234+5678+9ABC=

Uscita: 1234+5678+9ABC=DEFC

# CALCOLATORI ELETTRONICI

## Prova scritta

25 Settembre 1995

Scrivere un programma in Assembly che:

1. Accetti in ingresso una stringa di massimo 60 cifre decimali;
2. Accetti in ingresso una seconda stringa di 4 cifre decimali;
3. 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.
4. stampi la stringa ottenuta (la cui lunghezza deve essere uguale a quella della stringa in ingresso).

In un primo tempo assumere che la seconda stringa sia formata da 4 caratteri; in seguito estendere il programma al caso in cui la seconda stringa abbia lunghezza variabile (tra 2 e 9 caratteri).

Esempio:

Prima stringa in ingresso: 123412431434

Seconda stringa in ingresso: 1234

Uscita: 400121103000

# CALCOLATORI ELETTRONICI

## Prova scritta

29 Novembre 1995

Scrivere un programma in Assembly che:

1. Accetti in ingresso due date (giorno/mese/anno) nel formato: **gg/mm/aa**, dove l'intervallo di variabilità è:

**gg** varia da 01 a 30

**mm** varia da 01 a 12

**aa** varia da 50 a 99

Assumere che i dati inseriti soddisfino le specifiche elencate sopra.

2. calcoli e stampi il numero di giorni intercorsi tra le due date.

Assumere che la seconda data sia posteriore alla prima, che tutti i mesi siano formati da 30 giorni, che un anno sia formato da  $30*12=360$  giorni, e che non vi siano anni bisestili.

Successivamente considerare l'effettiva lunghezza dei mesi, e gli anni formati da 365 giorni.

Esempio:

Prima data: 01 12 94

Seconda data: 13 10 95

Giorni trascorsi: 312

# CALCOLATORI ELETTRONICI

## Prova scritta

19 Gennaio 1996

Scrivere un programma in Assembly che:

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

Determinare inoltre l'intervallo di variabilità delle ore (**hh**) in modo che tutti i calcoli possano essere effettuati utilizzando operazioni su 16 bit.

Assumere che il secondo orario sia posteriore al primo.

Successivamente 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

# CALCOLATORI ELETTRONICI

Prova scritta

19 Gennaio 1996

Scrivere un programma in Assembly che continui ciclicamente a stampare i numeri da 0 a 100 fintanto che non vengono premute opportune sequenze di tasti. In un primo tempo considerare l'unica sequenza di tasti possibile: "STOP" che termina il programma.

Successivamente considerare anche la sequenza "PAUSA" che ferma il ciclo fintanto che non viene premuto un tasto qualsiasi.

Al fine di controllare lo stato della tastiera utilizzare i servizi Interrupt numero 16h (servizi 0 e 1).

## CALCOLATORI ELETTRONICI

Prova scritta

27 Febbraio 1996

Scrivere un programma in Assembly che, introducendo un giorno e un mese dell'anno in corso (1996), determini il corrispondente giorno della settimana. Si ricorda che il 1 Gennaio 1996 è stato un lunedì.

Estendere in seguito il programma al calcolo del giorno della settimana di un qualunque giorno nel decennio 1990/1999. Si ricorda che il 1 Gennaio 1990 è stato un lunedì e che gli anni 1992 e 1996 sono anni bisestili.

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

Esempio per il primo caso:

Giorno: 27

Mese: 2

Il giorno della settimana e': Martedì

Esempio per il secondo caso:

Giorno: 27

Mese: 2

Anno: 1998

Il giorno della settimana e': Venerdì

# CALCOLATORI ELETTRONICI

## Prova scritta

25 giugno 1996

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  $80 \times 25$ ).

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  $(0, 80 - \text{lunghezza\_della\_stringa})$ ; nella successiva stampa della stringa tutti i caratteri devono essere stampati consecutivamente, eccetto l'ultimo, che deve essere stampato sotto il penultimo.

```
+-----+
| abc
|
|
|
|
+-----+
+-----+
| abc
|
|
|
|
+-----+
+-----+
| abc
|
|
|
|
+-----+
.
.
.
+-----+
|      ab|
|      c|
|
|
+-----+
+-----+
|      a|
|      b|
|      c|
|
+-----+
```

Il programma deve terminare dopo un giro completo della stringa.

Successivamente considerare la possibilità di uscire dal programma con la pressione del tasto 'SPAZIO' anche se la stringa non ha terminato il ciclo; al fine di controllare lo stato della tastiera utilizzare l'Interrupt numero 16h (servizi 0 e 1).

Successivamente consentire al programma di eseguire più di un ciclo: al termine di ogni ciclo la stringa verrà raddoppiata.

```
+-----+
|abcabc      |
|           |
|           |
|           |
+-----+
```

## ESEMPIO DI SOLUZIONE

```
dati      SEGMENT
coords    dw    ?
deltariga db    ?
deltacol  db    ?
strlen   dw    16
stringa  db    ' PROVA_DI_ESAME '
dati      ENDS

pila      SEGMENT STACK 'STACK'
          DB 1000 DUP (?)
pila      ENDS

ASSUME DS:dati,SS:pila,CS:codice

codice  SEGMENT PARA PUBLIC 'CODE'
main    proc far
        xor ax, ax
        push ds
        push ax
        call init
        mov cx, 80*2+20*2
scloop: mov dx, [coords]
        mov bh, [deltariga]
        mov bl, [deltacol]
        mov si, 0
lenloop: mov al, stringa[si]
        call printchar
        call inc_coords
        inc si
        cmp si, [strlen]
        jnz lenloop
        mov [deltariga], bh
        mov [deltacol], bl
        mov dx, [coords]
        call inc_coords
        mov [coords], dx
        call ritardo
        loop scloop
        ret
main    endp

ritardo proc near
        push cx
        mov cx, 30000
ciclo:  loop ciclo
        pop cx
ritardo endp

inc_coords proc near
        add dh, [deltariga]
        add dl, [deltacol]
        cmp dl, 80
        jnz no1ang
        dec dl
```

```

    inc dh
    mov [deltariga], 1
    mov [deltacol], 0
    jmp fine
no1ang:
    cmp dh, 21
    jnz no2ang
    dec dl
    dec dh
    mov [deltariga], 0
    mov [deltacol], OFFh
    jmp fine
no2ang:
    cmp dl, OFFh
    jnz no3ang
    inc dl
    dec dh
    mov [deltariga], OFFh
    mov [deltacol], 0
    jmp fine
no3ang:
    cmp dh, OFFh
    jnz fine
    inc dl
    inc dh
    mov [deltariga], 0
    mov [deltacol], 1
fine:   ret
inc_coords endp

printchar proc near
    push bx
    push ax
    mov bh, 0
    mov ah, 2
    int 10h
    pop ax
    push ax
    mov ah, 0Eh
    mov bl, 0
    int 10h
    pop ax
    pop bx
    ret
printchar endp

init    proc near
    mov ax, dati
    mov ds, ax
    mov [coords], 0
    mov [deltariga], 0
    mov [deltacol], 1
    ret
init    endp

codice ENDS
END main

```

# CALCOLATORI ELETTRONICI

## Prova scritta

16 luglio 1996

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.

In un primo tempo supporre che i dati inseriti soddisfino alle specifiche; in un secondo tempo effettuare il controllo degli errori.

# CALCOLATORI ELETTRONICI

## Prova scritta

10 Settembre 1996

Scrivere un programma in Assembly che

- accetti in ingresso una stringa di 10 caratteri formata esclusivamente 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. In più, 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.....

Ultima iterazione:

.....A...AA.....

.....S.....DS.....

.....B.....

## CALCOLATORI ELETTRONICI

## Prova scritta

1 Ottobre 1996

Scrivere un programma in Assembly che:

- 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 modifiche:
    - 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.

Successivamente introdurre il controllo sulla correttezza del formato della stringa inserita dall'utente.

Esempio:

aBcDefGhijkl  
abCdefGhijkl

# CALCOLATORI ELETTRONICI

## Prova scritta

19 Novembre 1996

Scrivere un programma in Assembly formato da un ciclo principale in cui un carattere viene mosso di una posizione per ogni iterazione. Il carattere partirà dall'angolo alto a sinistra dello schermo e si muoverà verso destra; una volta arrivato al bordo destro, scenderà di una posizione e continuerà a muoversi verso sinistra. Una volta arrivato al bordo sinistro, scenderà di una posizione e riprenderà secondo l'andamento descritto precedentemente. 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!

Al fine di controllare lo stato della tastiera si consiglia di utilizzare l'Interrupt numero 16h (servizi 0 e 1).

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;

# CALCOLATORI ELETTRONICI

## Prova scritta

28 Gennaio 1997

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; 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.

Successivamente:

- introdurre il controllo che la stampa non esca dallo spazio dello schermo; in tal caso segnalare errore.
- modificare il programma in modo che venga stampato 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  huja
dplpp
```

# CALCOLATORI ELETTRONICI

## Prova scritta

28 Febbraio 1997

Scrivere un programma in Assembly che:

- accetti in ingresso una stringa composta da massimo 70 caratteri; la stringa può contenere solo lettere maiuscole e minuscole;
- entri in un ciclo in cui ad ogni iterazione venga stampata una nuova stringa ottenuta dalla precedente eliminando le lettere (anche multiple) con codice ASCII minore.
- il ciclo termini quando la stringa risulta vuota.

Successivamente modificare il programma in modo che ad ogni iterazione vengano stampate due stringhe: la prima secondo quanto scritto sopra, mentre la seconda contenente le lettere eliminate dalla prima.

Esempio:

|                       |                   |                 |
|-----------------------|-------------------|-----------------|
| Inserire una stringa: | FFAFcAAMFFFEEEACf |                 |
| iterazione 1:         | FFFcMFFFEEEf      | AAAA            |
| iterazione 2:         | FFFMFfFEEEf       | AAAAC           |
| iterazione 3:         | FFFMFfF           | AAAACEEE        |
| iterazione 4:         | M                 | AAAAcCEEEFFFFf  |
| iterazione 5:         |                   | AAAAcCEEEFFFFfM |

# CALCOLATORI ELETTRONICI

## Prova scritta

2 Aprile 1997

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 parola 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
```

Inserire una stringa: Compito di Calcolatori Elettronici

```
*           *           *           *  
Calcolatori   Elettronici   Compito   Compito  
Calcolatori       Elettronici       di       Compito  
                           Calcolatori       Calcolatori  
                           Elettronici   Calcolatori  
                           Elettronici   Elettronici
```

# CALCOLATORI ELETTRONICI

## Prova scritta

7 Luglio 1997

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).

In un primo tempo assumere che il range di variabilità dell'istogramma sia tra 0 e 1000; successivamente determinare il range di variabilità utilizzando il valore massimo assunto dai dieci numeri, cioè tra 0 e  $\max(n_i)$ .

Esempio:

Inserire 10 numeri: 100, 200, 300, 100, 55, 45, 540, 230, 0, 200

|                             |         |
|-----------------------------|---------|
|                             | XXX     |
| XXX                         | XXX     |
| XXX                         | XXX XXX |
| XXX XXX                     | XXX XXX |
| XXX XXX XXX XXX XXX         | XXX XXX |
| XXX XXX XXX XXX XXX XXX XXX | XXX     |

# CALCOLATORI ELETTRONICI

## Prova scritta

10 Settembre 1997

Scrivere un programma in Assembly che:

- accetti in ingresso 2 stringhe alfanumeriche (che comprendono solo le lettere dell'alfabeto maiuscole e minuscole, e i numeri) senza spazi e con almeno un carattere in comune;
- stampi la prima stringa sullo schermo in direzione orizzontale da sinistra a destra;
- stampi la seconda stringa verticalmente dall'alto al basso in modo che le due stringhe si incrocino in corrispondenza del primo carattere che hanno in comune, che verrà sostituito da un asterisco.

In un primo tempo assumere che i dati in ingresso siano conformi alle specifiche. Successivamente introdurre il controllo sugli errori di inserimento.

Esempio:

Prima stringa: Albero  
Seconda stringa: REGALO

Output:

```
      R
      Alb*ro
      G
      A
      L
      O
```

## CALCOLATORI ELETTRONICI

## Prova scritta

12 Settembre 1997

Scrivere un programma in Assembly che permetta di inserire una stringa di lettere minuscole, maiuscole e numeri, di massimo 40 caratteri. Si supponga che la stringa contenga almeno 2 vocali. La prima e l'ultima vocale dividono la stringa in 3 parti.

Il programma deve pulire lo schermo e stampare la stringa iniziale in posizione centrale. Successivamente deve stampare su righe successive le tre stringhe in modo che la seconda stringa (la parte centrale della stringa iniziale) rimanga nella posizione originale, mentre le prima e la terza si spostino di un carattere verso sinistra e verso destra rispettivamente fino a raggiungere il bordo dello schermo. Quando entrambe le stringhe mobili hanno raggiunto il bordo dello schermo il programma termina.

### Successivamente

- introdurre i controlli sulla correttezza della stringa in ingresso e
  - fare in modo che l'insieme delle lettere che dividono la stringa iniziale sia definibile da utente.

Esempio:

```
Inserire la stringa: zxcvb3abcdefghijklmn
Inserire le lettere per dividerla: aeidfou

+-----+
|      zxcvb3abcdefghijklmn      |
|      zxcvb3 abcdefghi jklmn   |
|      zxcvb3 abcdefghi  jklmn  |
|      zxcvb3 abcdefghi  jklmn  |
|      zxcvb3 abcdefghi      jklmn |
|      zxcvb3 abcdefghi      jklmn|
|      zxcvb3 abcdefghi      jklmn|
|      zxcvb3 abcdefghi      jklmn|
|      zxcvb3 abcdefghi      jklmn|
+-----+
```

# CALCOLATORI ELETTRONICI

## Prova scritta

2 Dicembre 1997

Scrivere un programma in Assembly che permetta di inserire una stringa formata solo da lettere maiuscole e minuscole e che la visualizzi sullo schermo. Il programma deve poi entrare in un ciclo da cui può uscire solo alla pressione del tasto 'Spazio'. All'interno del ciclo viene controllata la tastiera; ogni volta che viene premuto il tasto corrispondente ad una lettera, dalla stringa iniziale vengono eliminate tutte le ricorrenze (maiuscole e minuscole) della 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')  
bcbCCBcdE

(dopo la pressione di 'b')  
cCCcdE

(dopo la pressione di 'c')  
dE

(dopo la pressione di 'B')  
bbBBdE

(dopo la pressione di 'A')  
abAAbaBBdE

(dopo la pressione di 'C')  
abcAAbCCaBBCdE

# CALCOLATORI ELETTRONICI

## Prova scritta

27 Gennaio 1998

Scrivere un programma in Assembly che permetta di inserire tre stringhe *a*, *b*, e *c* di caratteri alfanumerici (lettere e numeri) e le visualizzi sullo schermo. Il programma deve stampare la stringa *a*, carattere per carattere, partendo dalla metà del bordo sinistro dello schermo verso destra; ogni volta che si incontra una ricorrenza di un carattere della stringa *b*, la stampa procede verso l'alto di un carattere. Parallelamente ogni volta che si incontra una ricorrenza di un carattere della stringa *c*, la stampa procede verso il basso di un carattere.

Successivamente modificare il programma in modo che richieda l'immissione di una quarta stringa *d*; ogni volta che si incontra una ricorrenza di un carattere di quest'ultima stringa, la stampa della stringa *a* viene riportata nella posizione originale.

In un primo tempo si assuma che gli insiemi dei caratteri delle stringhe *b*, *c* e *d* abbiano intersezione nulla; successivamente introdurre tale controllo e, in caso di intersezione non nulla, mostrare le lettere in comune.

Esempio:

|                              |                      |
|------------------------------|----------------------|
| Inserire la prima stringa:   | zxccb3aqcdefghijklmn |
| Inserire la seconda stringa: | vb                   |
| Inserire la terza stringa:   | hl                   |
| Inserire la quarta stringa:  | f                    |

```
3aqcdef
b
zxccv      gh
          i j k l
          mn
```

# CALCOLATORI ELETTRONICI

## Prova scritta

4 Maggio 1998

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 si consideri che 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** **Nel mezzo del cammin di nostra vita**

# CALCOLATORI ELETTRONICI

## Prova scritta

23 Giugno 1998

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 abcedfgh AB abc xyz XYX

parole di lunghezza pari: abcedfgh AB

parole di lunghezza dispari: abc ABCDE abc xyz XYX

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

# CALCOLATORI ELETTRONICI

## Prova scritta

14 Luglio 1998

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 *maggior 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;
- non accettare il numero in tutte le altre situazioni.

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

Successivamente modificare il programma in modo che si comporti nel modo seguente:

- se la somma dei due ultimi numeri è pari, deve accettare il numero solo se esso è *strettamente maggiore* al precedente;
- se la somma dei due ultimi numeri è dispari, deve accettare il numero solo se esso è *strettamente minore* al precedente;
- non accettare il numero in tutte le altre situazioni.

Il programma deve uscire -oltre che dopo aver inserito il ventesimo carattere- anche se non è possibile inserire nessun numero che soddisfi le specifiche.

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

# CALCOLATORI ELETTRONICI

## Prova scritta

15 Settembre 1998

Scrivere un programma assembly 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  $5 \times 5$ , stampare questa matrice, e far evolvere gli elementi come descritto nel seguito:

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

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

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

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
```

# CALCOLATORI ELETTRONICI

## Prova scritta

23 febbraio 1999

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 a:

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

# CALCOLATORI ELETTRONICI

## Prova scritta

22 Giugno 1999

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: nelmezzodelcammindinostravita  
miritrovaiinunaselvaoscura  
Seconda stringa: 24578

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

# CALCOLATORI ELETTRONICI

## Prova scritta

13 Luglio 1999

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

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

N.B. Per semplificare il problema si assuma che due vertici del triangolo abbiano la stessa ordinata.

# CALCOLATORI ELETTRONICI

## Prova scritta

7 Settembre 1999

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 in modo decrescente rispetto al numero 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);
- ristampare le parole alternando consonanti e vocali a partire dalla prima consonante; in un primo tempo considerare parole formate da un ugual numero di consonanti e vocali, successivamente ripetere l'ultima vocale (se le vocali sono in numero minore) o consonante (se le consonanti sono in numero minore) per un numero di volte tale da rendere uguali il numero di vocali e il numero di consonanti nella parola.

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 lttrnc\*Eeoia nfrmfc\*Ioia nggnr\*Ieeia Crs\*oo dll\*ee Lr\*auea d\*i n\*i

Telecomunicazioni Elettronica Informatica Ingegneria Corso delle Lauree di in