

# Esercizi

## Nota Introduttiva

Gli esercizi che seguono (alcuni dei quali messi a punto con la collaborazione dell'Ing. Massimo Piotto, che ringrazio per la sua disponibilità) sono stati divisi in gruppi per quanto possibile omogenei in relazione al tipo di dati trattati o alle strutture di programmazione che richiedono. Per ogni gruppo sono stati a loro volta disposti (per quanto possibile) in ordine di difficoltà crescente.

Nel seguito si intende per “programmatore” la persona che realizza il VI e per “utente” quella che lo utilizza; l'utente non è necessariamente a conoscenza di come il programma (VI) è stato realizzato, ma deve essere messo in condizioni di impiegarlo correttamente.

Il pannello frontale (p. f.) è l'interfaccia utente dello strumento virtuale (VI). Attraverso il pannello frontale l'utente interagisce con il programma (o strumento virtuale), fornendo i dati in ingresso e prendendo visione di quelli in uscita.

I file .TXT o .DAT indicati con il testo di alcuni esercizi contengono i dati necessari all'esecuzione del programma<sup>1</sup>. Nel caso di esercizi di cui si propone uno (o, meglio, un possibile) svolgimento, è fornito anche il nome del VI corrispondente.

I riferimenti indicati accanto al titolo di alcuni paragrafi (es.: pag. 8-7) sono relativi al manuale *LabVIEW Fundamentals, Version 8.0*, reperibile sul sito web <http://www.ni.com>.

(vers. 1.0 - dicembre 2009)

## 1 Operazioni elementari su numeri interi e reali.

**1.1** - Scrivere in programma LabVIEW che, ricevuto in ingresso un valore numerico che rappresenta una temperatura espressa in gradi centigradi, lo converte nel corrispondente valore espresso in Kelvin.

**1.2** - Scrivere in programma LabVIEW che, ricevuto in ingresso un valore numerico intero che rappresenta l'ampiezza di un angolo espressa in gradi, lo converte nel corrispondente valore espresso in radianti. [es\_01\_02.vi]

**1.3** - Scrivere in programma LabVIEW che, ricevuto in ingresso un valore numerico che rappresenta l'ampiezza di un angolo espressa in radianti, lo converte nel corrispondente valore espresso in gradi, con arrotondamento del valore risultante ai centesimi di grado. [es\_01\_03.vi]

---

<sup>1</sup>In tutti i file contenenti dati numerici (non interi), il separatore decimale utilizzato è il punto; per utilizzare tali file su sistemi nei quali il separatore predefinito è la virgola, sarà in generale necessaria una sostituzione globale del punto con la virgola.

**1.4** - Scrivere in programma LabVIEW che, ricevuto in ingresso un valore numerico che rappresenta l'ampiezza di un angolo espressa in radianti, lo converte nel corrispondente valore espresso in gradi, primi e secondi, arrotondando il valore dei secondi ai centesimi. [es\_01\_04.vi]

**1.5** - Scrivere in programma LabVIEW che, ricevuto in ingresso un valore numerico che rappresenta l'ampiezza di un angolo espressa in radianti, lo converte nel corrispondente valore espresso in gradi, primi, secondi e centesimi di secondo, trascurando le cifre ulteriori (convertire prima il dato in centesimi di secondo e utilizzare quindi il nodo "quoziente e resto"). [es\_01\_05.vi]

**1.6** - Con riferimento a due segnali sinusoidali a frequenza  $f$  sfasati di un angolo  $\theta$  (espresso in radianti), scrivere un programma LabVIEW che legge il valore di  $f$  e di  $\theta$  e fornisce il valore del ritardo temporale  $\Delta t$  tra i due segnali (uno sfasamento positivo va inteso come ritardo del secondo segnale rispetto al primo). [es\_01\_06.vi]

**1.7** - Come Esercizio 1.6, ma leggendo  $f$  e  $\Delta t$  e fornendo in uscita il valore dello sfasamento: in caso di ritardo superiore a un periodo, lo sfasamento deve essere riportato all'interno dell'intervallo  $(0, 2\pi)$ . [es\_01\_07.vi]

**1.8** - Come Esercizio 1.7, ma riportando lo sfasamento all'interno dell'intervallo  $(-\pi, \pi)$ . [es\_01\_08.vi]

**1.9** - Scrivere in programma LabVIEW che, ricevuti i numeri  $a, b$  e  $c$ , calcola e visualizza su p.f. le radici dell'equazione  $ax^2 + bx + c = 0$ , accendendo opportuni indicatori luminosi per distinguere il caso di radici reali da quello di radici complesse e visualizzando in questo caso parte reale e coefficiente dell'immaginario. [es\_01\_09.vi]

**1.10** - Calcolare la somma accumulata dopo  $N$  anni a partire da un capitale iniziale  $x_0$  depositato in banca con interesse percentuale annuo  $i$  (es.:  $i = 5\%$ ); dopo ogni anno l'interesse maturato va a incrementare il capitale (interesse composto). [es\_01\_10.vi]

## 2 Manipolazione di stringhe

**2.1** - Fornire al programma LabVIEW, in ingresso, il valore di un numero reale  $x$  e stampare su pannello frontale la stringa  $\mathbf{x}$  = seguita dal valore della variabile  $x$  (arrotondata ai centesimi). [es\_02\_01.vi]

**2.2** - Data una coppia di numeri interi  $a$  e  $b$ , visualizzare su p.f. le stringhe "il quoziente è" e "il resto è" seguite, rispettivamente, dai valori del quoziente e del resto della **divisione intera** di  $a$  per  $b$ . p. es., per  $a = 10$  e  $b = 3$  il risultato da ottenere è:

il quoziente è 3

il resto è 1

[es\_02\_02.vi]

### 3 Cicli for/while (pag. 8-2, 8-3)

3.1 - Visualizzare su p.f., in sequenza e al ritmo di uno al secondo, i numeri interi da 1 a 10. [es\_03\_01.vi]

3.2 - Visualizzare su p.f., in sequenza e al ritmo di uno al secondo, i numeri interi da 1 a 10 e contemporaneamente, per ciascuno di essi, la relativa radice quadrata. [es\_03\_02.vi]

3.3 - Come Esercizio 3.2, ma con il numero di iterazioni e il tempo di attesa (espresso in secondi) impostabili da pannello frontale. [es\_03\_03.vi]

3.4 - Calcolare e visualizzare su p.f. la somma dei primi  $N$  multipli di un numero intero  $M$  ( $N$  ed  $M$  forniti da p.f.).

3.5 - Leggere da pannello frontale due interi  $n_1$  ed  $n_2$  ( $n_2 > n_1$ ) e visualizzare su p.f. al ritmo di uno al secondo, i numeri interi da  $n_1$  a  $n_2$  (estremi compresi). [es\_03\_05.vi]

3.6 - Leggere da pannello frontale due interi  $n_1$  ed  $n_2$  ( $n_2 > n_1$ ) e visualizzare su p.f., al ritmo di uno al secondo, i numeri da  $n_1$  a  $n_2$  (estremi compresi) con passo 0.5. [es\_03\_06.vi]

3.7 - Visualizzare su p.f., in sequenza e al ritmo di uno al secondo, il quadrato dei numeri interi  $i$  da 1 in poi, arrestando la visualizzazione dopo aver raggiunto (e visualizzato il corrispondente quadrato) il minimo intero per cui si ha

$$m = i^2 - 2 * i > 20$$

(visualizzare anche il valore di  $i^2$  e quello di  $m$ ). [es\_03\_07.vi]

3.8 - Visualizzare su p.f. in sequenza e al ritmo di uno al secondo, i primi  $n$  numeri dispari (il valore di  $n$  è fornito da pannello frontale). [es\_03\_08.vi]

3.9 - Visualizzare su p.f., al ritmo di uno al secondo, i numeri interi ( $i$ ) da 0 a 30; sul p.f. è accesa una spia verde se  $i \leq 15$ ; in caso contrario la spia è spenta e ad ogni incremento dell'indice viene prodotto un segnale sonoro. [es\_03\_09.vi]

#### Shift Register in cicli for (pag. 8-7)

3.10 - Calcolare e visualizzare su p.f. la somma dei primi  $n$  numeri interi (il valore di  $n$  viene inserito da p.f.). [es\_03\_10.vi]

3.11 - Utilizzando un ciclo for e uno shift register visualizzare su p.f. in sequenza (al ritmo di uno al secondo), gli elementi della successione

$$A_0 = 0, \quad A_i = A_{i-1} + \sqrt{i} \quad \text{per } i \leq 10.$$

[es\_03\_11.vi]

**3.12** - Calcolare e visualizzare su p.f. la somma dei primi  $n$  numeri dispari (il valore di  $n$  è fornito da pannello frontale). [es\_03\_12.vi]

**3.13** - Calcolare e visualizzare su p.f. il fattoriale di un numero  $N$  (fornito da p.f.), senza utilizzare la funzione LabVIEW `fact.vi`. [es\_03\_13.vi]

**3.14** - Calcolare e visualizzare su p.f., in sequenza e al ritmo di un valore al secondo, la radice quadrata della somma dei quadrati dei primi  $1, 2, \dots, n$  numeri interi (il valore di  $n$  viene inserito da p.f.). Dunque inserendo il valore 3 da pannello frontale si otterrà la visualizzazione, in sequenza, dei numeri  $\sqrt{1^2}(= 1)$ ,  $\sqrt{1^2 + 2^2}(= \sqrt{5} \simeq 2.24)$  e  $\sqrt{1^2 + 2^2 + 3^2}(= \sqrt{14} \simeq 3.74)$ . [es\_03\_14.vi]

**3.15** - Calcolare e visualizzare su p.f., in sequenza e al ritmo di uno al secondo, la radice quadrata della somma dei quadrati dei primi  $1, 2, \dots, n$  numeri (interi) dispari (il valore di  $n$  viene inserito da p.f.). Dunque inserendo il valore 3 da pannello frontale si otterrà la visualizzazione, in sequenza, dei numeri  $\sqrt{1^2}(= 1)$ ,  $\sqrt{1^2 + 3^2}(= \sqrt{10} \simeq 3.16)$  e  $\sqrt{1^2 + 3^2 + 5^2}(= \sqrt{35} \simeq 5.92)$ . [es\_03\_15.vi]

**3.16** - Calcolare il seno di un angolo  $x$  (espresso in radianti) utilizzando lo sviluppo in serie di Taylor (nell'intorno di  $x = 0$ ) arrestato al termine di ordine  $N$ :

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} \dots$$

con  $x$  ed  $N$  forniti da p.f., e visualizzarlo su p.f.; se  $N$  è pari lo sviluppo andrà arrestato al termine di grado  $N - 1$ . Confrontare il risultato con il valore effettivo di  $\sin(x)$ , visualizzando anche questo su p.f. (per il calcolo del fattoriale v. Esercizio 3.13). [es\_03\_16.vi]

## 4 Vettori

**4.1** - Utilizzando un ciclo `for`, costruire il vettore  $A_1 \dots A_{10}$  di 10 elementi  $A_i = i^2$ , (ogni elemento del vettore è il quadrato del corrispondente indice) e visualizzarlo (alla fine del ciclo `for`) sul pannello frontale; durante la costruzione del vettore visualizzare l'indice e il corrispondente quadrato (per rendere leggibile la visualizzazione durante la costruzione del vettore inserire un'opportuna attesa). [es\_04\_01.vi]

**4.2** - Calcolare gli elementi di un array di  $n$  elementi  $a_i = 1/(2i + 1)$  ( $n$  fornito da p.f.,  $1 < i < n$ ). Visualizzare l'array su p.f. durante la sua costruzione. [es\_04\_02.vi]

**4.3** - Visualizzare su p.f., in sequenza crescente e al ritmo di uno al secondo, i numeri interi e le rispettive radici quadrate, arrestando la successione quando la radice quadrata risulta maggiore di (o uguale a) un valore  $x$  fornito da p.f. Quindi visualizzare il vettore i cui elementi sono le radici quadrate calcolate fino a questo punto. [es\_04\_03.vi]

**4.4** - Ripetere l'esercizio 3.11 memorizzando i valori  $A_i$  in un vettore; visualizzare anche quest'ultimo su p.f. [es\_04\_04.vi]

**4.5** - Calcolare gli elementi di un array bidimensionale ( $n$  righe ed  $m$  colonne, con  $n$  ed  $m$  forniti da p.f.)  $a_{i,j} = 1/(i + 2 * j + 1)$ ;  $i$  è l'indice di riga (ovviamente con  $1 < i \leq n$ ) e  $j$  l'indice di colonna ( $1 < j \leq m$ ); visualizzare l'array su p.f. [es\_04\_05.vi]

### **Auto indicizzazione/for (pagg. 8-5 e 8-6)**

**4.6** - Calcolare la media di un insieme di numeri, forniti come elementi di un array. (L'array viene fornito in ingresso da pannello frontale). [es\_04\_06.vi]

**4.7** - Scrivere un programma che legge una serie di valori numerici (da p. f.), ne calcola la media, la differenza di ciascun valore rispetto alla media e la radice quadrata dello scarto quadratico medio. [es\_04\_07.vi]

**4.8** - Calcolare la media ponderata di un insieme di numeri, forniti come elementi di un array; i relativi pesi sono gli elementi di un secondo array. (Gli array vengono forniti in ingresso da pannello frontale). [es\_04\_08.vi]

**4.9** - Calcolare il prodotto cumulativo di una serie di numeri, forniti come elementi di un array. (L'array viene fornito in ingresso da pannello frontale). [es\_04\_09.vi]

**4.10** - Calcolare la media geometrica di una serie di numeri, forniti come elementi di un array. (L'array viene fornito in ingresso da pannello frontale). [es\_04\_10.vi]

**4.11** - Scrivere un programma che legge una serie di  $n$  valori numerici da pannello frontale, verifica se  $n \geq 5$  e, in caso affermativo, restituisce la somma dei primi  $n - 5$  termini; in caso contrario segnala (con un indicatore rosso su p.f.) la condizione  $n < 5$  e fornisce il risultato 9999 su p.f.; l'indicatore rimane spento se  $n \geq 5$ . [es\_04\_11.vi]

**4.12** - Generare, al ritmo di 1 al secondo, i primi  $N$  termini della successione di Fibonacci. Durante l'esecuzione del programma vengono visualizzati l'indice (0, 1, . . . N-1) e il relativo elemento; alla fine dell'esecuzione viene visualizzato un vettore contenente gli  $N$  termini generati. Il valore di  $N$  viene immesso da p.f.. [es\_04\_12.vi]

## **5 Grafici e diagrammi**

**5.1** - Utilizzando una *Waveform Graph*, rappresentare gli elementi del vettore dell'Esercizio 4.1 (in funzione dell'indice  $i$ ). [es\_05\_01.vi]

**5.2** - Ripetere l'esercizio 5.1, ma utilizzando un *x-y Graph*. [es\_05\_02.vi]

**5.3** - Assegnare ai 100 elementi di un vettore  $A$  i valori assunti da una senoide (di ampiezza unitaria) in altrettanti istanti (equispaziati) all'interno di un periodo. Rappresentare, utilizzando un *Waveform Graph*, l'andamento della funzione al variare dell'angolo  $\theta$  ( $0 \leq \theta \leq 2\pi$ ).

[es\_05\_03.vi]

**5.4** - Assegnare ai 100 elementi di un vettore  $A$  i valori assunti da una senoide (di ampiezza unitaria) in altrettanti istanti (equispaziati) all'interno di un periodo e a un vettore  $B$  i corrispondenti valori di una cosenoide di ampiezza doppia. Rappresentare, utilizzando un *Waveform Graph*, l'andamento delle due funzioni al variare dell'angolo  $\theta$  ( $0 \leq \theta \leq 2\pi$ ).

[es\_05\_04.vi]

**5.5** - Ripetere l'esercizio 5.4, ma utilizzando un *x-y Graph*.

[es\_05\_05.vi]

## 6 Lettura da file/scrittura su file

**6.1** - Leggere da un file di testo un vettore di dati interi (organizzati in unica riga) e visualizzarlo su pannello frontale.

[dati\_riga.dat, es\_06\_01.vi]

**6.2** - Leggere da un file un vettore di dati interi (organizzati in unica colonna) e visualizzarlo sul pannello frontale.

[dati\_colonna.dat, es\_06\_02.vi]

**6.3** - Leggere da un file un vettore di dati interi (con la possibilità di selezionare da p. f. il formato (riga o colonna) dei dati nel file) e visualizzarlo su pannello.

[dati\_riga.dat, dati\_colonna.dat, es\_06\_03.vi]

**6.4** - Leggere da disco il file `dati_colonna.dat` contenente un vettore colonna di numeri interi e visualizzare l'intero vettore su pannello frontale; per ciascun elemento del vettore visualizzare quindi il relativo quadrato (uno per volta e al ritmo di uno ogni 0.5 s).

[es\_06\_04.vi]

**6.5** - Come Esercizio 6.3, con in aggiunta la visualizzazione in sequenza, dal primo all'ultimo e con un ritardo di 2 s tra l'uno e l'altro, degli elementi del vettore e del relativo indice.

[es\_06\_05.vi]

**6.6** - Come Esercizio 6.5, ma visualizzando gli elementi del vettore in ordine inverso, cioè dall'ultimo al primo.

[es\_06\_06.vi]

**6.7** - Lettura da un file *spreadsheet* di un vettore di  $N$  dati organizzati in unica colonna e successiva visualizzazione su pannello frontale dell'intero vettore (come Esercizio 6.1); in aggiunta a questo, sul pannello frontale dovranno essere visualizzate, una ogni 2 secondi, le coppie di elementi:

$$\begin{aligned} a(i) \text{ e } a(i+1) & \text{ per } 1 \leq i \leq N-1 \\ a(i) \text{ e } 0 & \text{ per } i = N \end{aligned}$$

[es\_06\_07.vi, dati\_colonna.dat]

- 6.8** - Salvare su file *spreadsheet* un array di 10 numeri generati casualmente disponendoli tutti sulla stessa riga, separati da un carattere di tabulazione; (visualizzare l'array su p.f.).  
[es\_06\_08.vi]
- 6.9** - Salvare su file *spreadsheet* un array di 10 numeri generati casualmente disponendoli tutti sulla stessa riga, separati dal carattere ' ; '.  
[es\_06\_09.vi]
- 6.10** - Salvare su file *spreadsheet* un array di  $n$  numeri generati casualmente (con  $n$  letto da pannello frontale) disponendoli uno per riga.  
[es\_06\_10.vi]
- 6.11** - Simulare il lancio di un dado ripetuto 10 volte (generando ogni volta un numero casuale e riportando il valore ottenuto sull'intervallo di interi (1...6)); visualizzare l'array di valori (interi) ottenuti su p.f. e salvarlo quindi su file (un numero **intero** per ogni riga).  
[es\_06\_11.vi]
- 6.12** - Leggere da file (*stud.txt*) un vettore di stringhe che rappresentano i nomi di  $n$  studenti e (*voti.dat*) un vettore  $n \times m$  ( $n$  righe,  $m$  colonne) di numeri interi compresi tra 0 e 10 che rappresentano i voti riportati dagli studenti in  $m$  diverse materie scolastiche. Fornire in uscita, per ciascuno studente, la media dei relativi voti e un indicatore luminoso che segnala con il suo colore se la media è sufficiente (verde) o insufficiente (rosso).  
[es\_06\_12.vi]
- 6.13** - Scrivere un programma LabVIEW che, letto da file un array bidimensionale  $n \times m$  ( $n$  righe,  $m$  colonne) di numeri reali e ricevuti in ingresso (da p. f.) due interi  $i$  e  $j$  ( $1 < i < n$  e  $1 < j < m$ ), produce la visualizzazione su p.f. della matrice letta, della riga  $i$ -esima e della colonna  $j$ -esima.  
[es\_06\_13.vi, tab\_reali.dat]
- 6.14** - Ripetere gli Esercizi 6.1 e 6.2, con l'aggiunta della visualizzazione su diagramma cartesiano (*Waveform Graph*) degli elementi del vettore in funzione dell'indice.
- 6.15** - Ripetere l'Esercizio 6.14, con l'aggiunta della visualizzazione sullo stesso diagramma anche dei quadrati degli elementi del vettore in funzione dell'indice.
- 6.16** - Leggere da file un array bidimensionale quadrato ( $n \times n$ ) di numeri e visualizzarlo su p.f.; visualizzare inoltre, in sequenza e con la cadenza di 1 s, i termini sulla diagonale principale.  
[es\_06\_16, tab\_quad.dat]
- 6.17** - Leggere da file un array bidimensionale quadrato  $n \times n$  di numeri, visualizzarlo su p.f. e fornire in uscita il prodotto dei termini sulla diagonale principale. [dati: tab\_quad.dat]
- 6.18** - Leggere da file un array bidimensionale quadrato  $n \times n$  di numeri, visualizzarlo su p.f. e fornire in uscita il prodotto dei termini sulla diagonale secondaria. [es\_06\_18, tab\_quad.dat]

## 7 Miscellanea

7.1 - Scrivere un programma LabVIEW che legge da p.f. un numero reale  $x$  e visualizza:

1 se  $x \leq 1$   
2 se  $1 < x \leq 2$   
3 se  $2 < x \leq 3$   
99 altrimenti

[es\_07\_01.vi]

7.2 - Simulare il lancio di un dado (v. Esercizio 6.11) ripetuto 1000 volte; produrre quindi un diagramma delle frequenze (cioè un diagramma in cui, per ogni valore da 1 a 6, è riportato il rapporto tra il numero di volte in cui quel numero è stato ottenuto come risultato e il numero totale dei lanci). Riportare sullo stesso diagramma l'andamento teorico della frequenza (densità di probabilità).

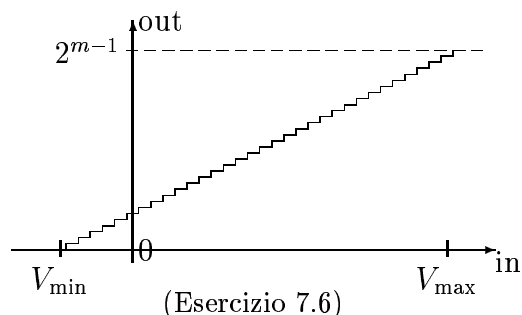
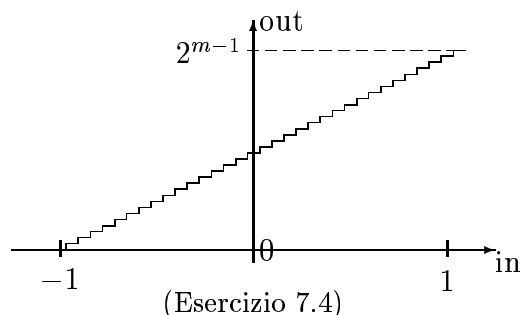
[es\_07\_02.vi]

7.3 - Generare un array di numeri reali, i cui  $2^n$  elementi (con  $n$  fornito da p.f.) sono i campioni di un periodo di un segnale sinusoidale di valore di picco unitario. Visualizzare l'andamento della funzione come *Waveform Graph*.

[es\_07\_03.vi]

7.4 - Digitalizzare il segnale dell'esercizio 7.3, su  $m$  bit di risoluzione verticale, convertendo ciascun campione in un numero intero non negativo, secondo la caratteristica ingresso-uscita riportata in figura. Visualizzare il risultato di questa operazione come *Waveform Graph*.

[es\_07\_04.vi]



7.5 - Ripetere gli Esercizi 7.3 e 7.4 con un segnale a onda triangolare a valor medio nullo e valore di picco unitario.

7.6 - Leggere da un file *spreadsheet* i campioni (relativi a un periodo) di un segnale e visualizzarne su p.f. l'andamento. Quindi digitalizzarlo su  $2^m$  livelli di risoluzione verticale (come nell'esercizio 7.4), impiegando una caratteristica che si adatta perfettamente alla dinamica  $[V_{\min} \div V_{\max}]$  del segnale utilizzato, i cui valori minimo ( $V_{\min}$ ) e massimo ( $V_{\max}$ ) non sono noti *a priori* e vanno dunque calcolati (v. figura). Visualizzare il risultato della digitalizzazione come *Waveform Graph*.

[es\_07\_06.vi, sincos.dat]



**7.7** - Con riferimento a un filtro di Butterworth di ordine  $n$  realizzato con celle di Sallen e Key, scrivere un programma che legge il valore di  $n$  e fornisce su p.f. il numero di celle necessarie, il valore (espresso in radianti e in gradi) degli angoli caratteristici dei poli del filtro, il valore del guadagno ( $A_v$ ) di ciascuna cella, il corrispondente coefficiente del termine di primo grado del polinomio caratteristico della cella ( $3 - A_v$ ) e indica su p.f. se è necessario aggiungere alle celle di S. e K. anche uno stadio del primo ordine.

Ad esempio, per  $n = 3$  vengono forniti gli angoli  $\pi$  ( $180^\circ$ ),  $2\pi/3$  ( $120^\circ$ ),  $4\pi/3$  ( $240^\circ$ ), il guadagno  $A_v = 2$ , il coefficiente  $3 - A_v = 1$  e l'indicazione della necessità di uno stadio del primo ordine.

**7.8** - Scrivere un programma per disegnare le figure di Lissajous. L'utente inserisce da p.f. il numero  $N$  di punti per periodo, l'ampiezza, la frequenza e la fase dei due segnali (sinusoidali) e ottiene sul pannello frontale la figura di Lissajous relativa e due vettori contenenti i valori dei due segnali. [es\_07\_08.vi]

**7.9** - Come esercizio precedente, ma il numero di punti per periodo è fisso e pari a 1000 e l'utente fornisce da p.f. il rapporto  $n$  tra le frequenze dei due segnali, le due ampiezze e lo sfasamento reciproco. Sul pannello frontale si ottengono i vettori contenenti i valori dei campioni dei segnali, i relativi diagrammi temporali (*Waveform Graph*) e la figura di Lissajous corrispondente. [es\_07\_09.vi]

**7.10** - Si scriva un programma in grado di calcolare il fattoriale di un numero  $N$ . L'utente inserisce il numero  $N$  di cui vuole calcolare il fattoriale e il programma restituisce:

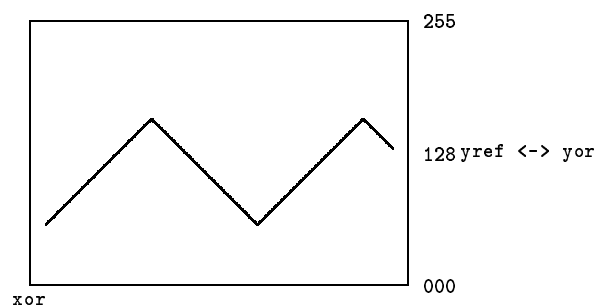
- il valore di  $N!$ ;
- un vettore contenente i fattoriali dei numeri compresi tra 0 ed  $N$ , estremi inclusi;
- un vettore in cui ogni elemento è costituito dall'espressione " $x!=<\text{fattoriale di } x>$ " (es.:  $3!=6$ ) per i numeri compresi tra 0 ed  $N$ , estremi inclusi.

[es\_07\_10.vi]

**7.11** - Nel file (di testo) `lettture.txt` è memorizzata la stringa derivante dalla lettura di  $n = 3$  valori di tensione con un unico impulso di trigger mediante un multimetro HP 34401A (:SAMP:COUNT 3). Leggere la stringa da file, visualizzarla su p.f., estrarre i 3 valori di tensione e visualizzarli su p.f. come array di numeri reali. [es\_07\_11.vi]

**7.12** - Come Esercizio 7.11 ma in modo che il programma funzioni correttamente senza che sia noto *a priori* il numero di letture effettuate (il valore di  $n$  deve essere ricavato durante l'esecuzione del programma). [es\_07\_12.vi]

**7.13** - Generare un array di valori di frequenza, in progressione geometrica, a partire dal valore  $f_0$ , con  $n$  valori per decade, e per  $m$  decadi; (aggiungere come ultimo elemento dell'array è il primo valore di frequenza della decade successiva); i valori di  $f_0$ ,  $n$  ed  $m$  vengono immessi da p.f.. [es\_07\_13.vi]



(Esercizio 7.14)

**7.14** - L'oscilloscopio digitale HP 54600, se interrogato tramite interfaccia remota, fornisce la traccia visualizzata nella forma di:

- una stringa di caratteri (detta *preambolo*) nel formato:

`<format>,<type>,<points>,<count>,<xinc>,<xor>,<xref>,<yinc>,<yor>,<yref>`

in cui:

- `<format>`, è un carattere (0 o 1 o 2) che rappresenta il formato dei dati (0=ASCII, 1=BYTE, 2=WORD);
- `<type>`, è un carattere (0 o 1 o 2) che rappresenta il tipo di acquisizione (0=AVERAGE, 1=NORMAL, 2=PEAK DET);
- `<points>`, rappresenta il numero di punti di misura letti;
- `<count>`, è un carattere (presente per motivi di compatibilità) privo di significato in questo contesto;
- `<xinc>`: è la distanza temporale (in secondi) tra un campione e il successivo;
- `<xor>`: è l'ascissa (in secondi) corrispondente al primo dato;
- `<xref>`: *time reference point*, indica il primo punto a sinistra sullo schermo (0);
- `<yinc>`: è la risoluzione verticale (in Volt);
- `<yor>`: è il livello di tensione corrispondente al livello di riferimento (a metà schermo);
- `<yref>`: è il valore (digitalizzato) corrispondente al livello di riferimento;
- una stringa di caratteri che rappresenta i dati (digitalizzati su  $n$  bit), costituita da
  - il carattere #;
  - un carattere che esprime il numero ( $m$ ) di caratteri seguenti dai quali andrà ricavato il numero di dati acquisiti;
  - gli  $m$  caratteri che indicano il numero ( $n$ ) di byte trasmessi;

- $n$  byte corrispondenti ai dati, rappresentati da numeri interi su tre cifre, compresi tra 0 e  $2^8 - 1 = 255$  (il convertitore A-D lavora su 8 bit);

$$\overbrace{\#8\ 00002000}^{\text{header}}\langle\text{dati}\rangle\langle\text{terminatore}\rangle$$
 8 caratteri

nella stringa mostrata come esempio, dopo il carattere #, è presente un 8, il quale indica che negli 8 caratteri successivi (00002000) è indicato il numero (2000) di byte inviati dall'oscilloscopio; i primi 10 caratteri costituiscono il cosiddetto *header*.

Tra ogni dato e il successivo c'è una virgola; non c'è la virgola *prima* del primo dato; l'ultimo dato è seguito da un terminatore ( $\langle\text{nl}\rangle$ ). Dunque il numero di byte inviati è pari al quadruplo dei valori numerici da leggere: per ogni valore vengono inviati 3 byte contenenti le 3 cifre, più la virgola (o il terminatore finale).

- I file `pre1.txt` e `data1.txt` contengono, rispettivamente, il preambolo e i dati relativi a una traccia visualizzata sull'oscilloscopio e digitalizzata; la stringa dei dati è stata privata dei caratteri che costituiscono l'*header*. Estrarre i dati dal relativo file (sapendo che sono in numero di 500), convertirli in valori di tensione e visualizzare la traccia corrispondente sul p.f..
- Ripetere l'esercizio precedente ricavando il numero di dati (per esempio) dalla lunghezza della stringa contenuta nel file `data1.txt`.
- Ripetere l'esercizio utilizzando i file `pre2.txt` e `data2.txt`; quest'ultimo file contiene anche l'*header*, dal quale va ricavato il numero di dati presenti.

Attenzione: l'istruzione *Scan from String* (Menu *String*), che consente di convertire porzioni di una stringa in numeri (interi, reali, ecc.) di cui sia noto *a priori* il formato, può essere utile per lo svolgimento di questo esercizio.

File dati:

- `dati_riga.dat` array monodimensionale di numeri interi, disposti su una riga e separati dal carattere di tabulazione;
- `dati_colonna.dat` array monodimensionale di numeri interi, disposti su una colonna;
- `stud.txt` array monodimensionale di stringhe (cognome e nome di  $n$  persone), disposte una per riga;
- `tab.dat`: array bidimensionale di numeri interi separati (sulla riga) dal carattere di tabulazione;
- `tab_reali.dat`: array bidimensionale di numeri reali separati (sulla riga) dal carattere di tabulazione;
- `tab_quad.dat`: array bidimensionale di numeri interi separati (sulla riga) dal carattere di tabulazione; il numero di righe è pari al numero di colonne;
- `tab1.dat`: array bidimensionale di numeri interi separati (sulla riga) dal carattere “punto e virgola”;
- `sincos.dat`: array monodimensionale di numeri reali, disposti su una riga e separati dal carattere di tabulazione; rappresentano i campioni di un periodo di forma d’onda;
- `voti.dat`, `letture.txt`, `pre1.txt`, `data1.txt`, `pre2.txt`, `data2.txt`: file per specifici esercizi.