

10 - ESERCIZI SULLE STRUTTURE ITERATIVE

Esercizio 9)

Riprendi il programma

```
program somma_di_un_numero_non_prefissato_di_addendi
```

(vedi paragrafo precedente)

e ribattezzalo

```
program prodotto_di_un_numero_non_prefissato_di_fattori
```

modificandolo affinché esegua, non più la somma, bensì il *prodotto* dei numeri letti da tastiera.

Esercizio 10)

Realizza un programma (program **margherita**), che faccia comparire sullo schermo il seguente output:

```
1 M'AMA
2 NON M'AMA
3 M'AMA
4 NON M'AMA
.....
n ...
```

essendo n un numero (pseudo)casuale compreso fra 1 e 20.

INDICAZIONI

Puoi usare la struttura iterativa che ritieni più opportuna, ma successivamente ti invito a rifare il programma altre due volte, utilizzando, come utile esercizio, le due strutture iterative rimanenti.

Ricordiamo che il “test di parità”

si effettua mediante l’operatore MOD (che dà il resto della divisione intera):

if a mod b = 0 significa: “**se a è divisibile per b**” e perciò:

if x mod 2 = 0 significa: “**se x è pari**”

if x mod 2 <> 0 oppure **if x mod 2 =1** significa: “**se x è dispari**”

Sarà opportuno inserire un “**ritardo**” fra un “petalo” e l’altro.

Vedi a questo proposito, al paragrafo 5g, “L’effetto ritardo”.

Esercizio 11)

Scrivi **un programma che, letta da tastiera una sequenza di numeri qualsiasi, scriva sullo schermo il minimo ed il massimo fra i numeri introdotti.**

Per indicare che la sequenza dei numeri in input è terminata, l’utente digiterà il numero 0 (“sentinella”).

Attenzione!

La “sentinella” 0 segnala la fine della sequenza,

ma *non fa parte* dell’insieme dei numeri di cui si desidera conoscere il massimo e il minimo.

Occhio perciò a metter giù il programma correttamente.

Converrà preliminarmente far leggere il primo numero della sequenza,

per assegnare come valore iniziale a ciascuna delle due variabili

massimoprovisorio e *minimoprovisorio*,

il valore del numero letto.

Esercizio 12)

Scrivi **un programma che si occupi di far fare al fratellino piccolo 10 esercizi sulle tabelline** (un po’ come nell’esercizio 3 del paragrafo 6).

Il computer, di fronte ad una risposta sbagliata,

deve continuare a riproporre la stessa moltiplicazione fino a quando il bambino risponde esattamente.

I fattori per ciascun esercizio dovranno essere (pseudo)casuali

(ciascun fattore potrà andare da 0 a 10).

Il programma dovrà anche contare il numero complessivo di risposte sbagliate.

Esercizio 13)

Letto in ingresso un intero n, il programma ne elenca i divisori e li conta, includendo anche i divisori “impropri” (che sono il numero stesso e l’unità).

Esercizio 13')

Completare il programma precedente, in modo che fornisca in output anche la scritta IL NUMERO E' PRIMO oppure IL NUMERO NON E' PRIMO, a seconda dei casi.

E' chiaro che la prima circostanza si verifica quando i divisori sono soltanto 2 (l'unità e il numero).

Esercizio 14)

Letto in ingresso un intero positivo n, stabilire se si tratta di un numero primo.

Questa volta, però, ti chiedo di **RIDURRE il numero di passi che la macchina dovrà compiere prima di fornire la risposta.**

A tale scopo, non si stanno a individuare e contare *tutti* i divisori di n ;

si prova invece a vedere se n è divisibile per 2, poi per 3, ecc., fermandosi quando

a) si trova un divisore

b) oppure (operatore logico OR) il numero che è "candidato" ad essere un divisore di n ha già superato $\text{sqrt}(n)$, ossia la radice quadrata di n

(infatti si dimostra che, se in tal caso non sono ancora stati trovati divisori propri, n non potrà avere alcun divisore proprio; i divisori "propri" sono quelli diversi dall'unità e dal numero stesso).

Esercizio 14')

Integra il programma precedente per far sì che in output venga fornita

la **sequenza di tutti i numeri primi non superiori a MAX, essendo MAX un intero letto in ingresso.**

Considera la possibilità di mandare i numeri ordinatamente in output senza andare a capo: ad es., se utilizzi **write(x:8)** l'intero x verrà scritto sul monitor allineandolo a destra su di un campo di 8 caratteri.

Tieni conto del fatto che **ogni riga della finestra FREE PASCAL contiene esattamente 80 caratteri.**

Variante: un programma che scriva la sequenza degli interi, coi soli numeri primi colorati in rosso.

Esercizio 15)

PREMESSA Si dice "fattoriale" di un intero $n > 0$, e lo si indica con **$n!$** (leggi: " **n fattoriale**"), il prodotto dei fattori interi decrescenti da n fino a 1:

$$n! = n \cdot (n-1) \cdot \dots \cdot 3 \cdot 2 \cdot 1. \text{ Ad esempio: } 4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

Scrivi un programma che, letto in input n , calcoli $n!$

Esercizio 16)

Gara di testa-e-croce.

Ricordiamo che un'istruzione del tipo `moneta:=random(2)` ha l'effetto di assegnare alla variabile "moneta" (di tipo intero) il valore 0 oppure il valore 1, in modo (pseudo)casuale.

Se interpretiamo, ad esempio, il valore 0 come "testa" e il valore 1 come "croce",

avremo così "simulato" il lancio di una moneta.

Si lancia dunque per n volte una moneta (con n letto in ingresso)

e si vuole stabilire quante volte è uscita "testa" (cioè, 0) e quante volte "croce" (cioè, 1).

E' opportuno far sì che la sequenza di "colpi" compaia in output.

Esercizio 17)

Si lancia un dado più volte; ci si ferma dopo che è uscito per esattamente 100 volte il numero 6.

Si vuole sapere quanti lanci sono stati effettuati in totale.

Esercizio 18)

Scrivi un programma che, letti in ingresso due numeri interi a, b , ne calcoli il **minimo comune multiplo**.

Ci sono diversi modi alternativi per impostare l'algoritmo;

ad esempio, si può calcolare la successione dei multipli di a ($a, 2a, 3a, 4a, \dots$), arrendendosi

quando si perviene ad un numero che risulta multiplo anche di b (cioè, che risulta divisibile per b).

Ricorda che in PASCAL l'asterisco di moltiplicazione NON può essere lasciato sottinteso.

Esercizio 19)

Scrivi un programma che, letti in ingresso due numeri interi a, b , ne calcoli il **massimo comun divisore**.

Puoi utilizzare **diversi metodi alternativi**, fra cui l' "algoritmo di Euclide". Scegli quello che desideri.

Esercizio 20)

Dato un intero $n > 1$, scomporlo in fattori primi.

Esercizio 21)

Letti in input i due termini a, b di una frazione, ridurre la frazione data ai minimi termini.

Ad esempio, se l'input è

180	l'output dovrà essere	2
450		5