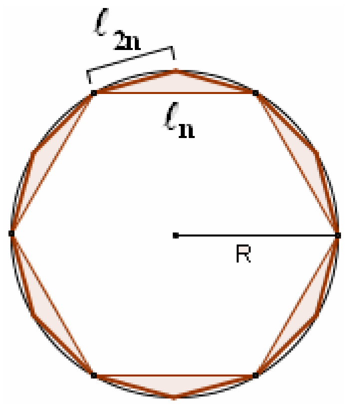
12 - APPROSSIMAZIONI DI PI GRECO



Se indichiamo con ℓ_n la misura del lato del poligono regolare di n lati, inscritto nella circonferenza di raggio R,
allora si dimostra che la misura del lato del poligono regolare, inscritto nella stessa circonferenza, ma avente numero di lati DOPPIO, è dato da

$$\boxed{\ell_{2n} = \sqrt{2R^2 - R\sqrt{4R^2 - \ell_n^{\,2}}} \qquad \textbf{FORMULA } \ell_n \to \ell_{2n}}$$

Poniamo ora per semplicità R=1; la formula diventa $\boxed{\ell_{2n} = \sqrt{2 - \sqrt{4 - \ell_n^{\,2}}}}$

Se partiamo dall'esagono regolare inscritto ($\ell_6 = 1$: è noto che il lato dell'esagono regolare inscritto è uguale al raggio della circonferenza), applicando ripetutamente questa formula potremo calcolare ℓ_{12}, ℓ_{24}, ℓ_{48} ...

Osserviamo ora che i perimetri dei corrispondenti poligoni regolari inscritti costituiscono approssimazioni per difetto, sempre più precise, della lunghezza della circonferenza.

Ma tale lunghezza è data da $2\pi \cdot raggio = 2\pi$ (ricordiamo che abbiamo preso R=1),
per cui i SEMIperimetri dei poligoni regolari considerati
costituiranno approssimazioni per difetto, sempre più precise, del numero π.

Esercizio 39)
Scrivi un programma Pascal che, a partire dal lato dell'esagono regolare inscritto nella circonferenza di raggio R=1 ($\ell_6 = R = 1$), calcoli successivamente la misura del lato del poligono regolare inscritto avente: 12, 24, 48, 96, 192 … lati,
e mandi in output tale misura insieme con la misura del SEMIperimetro del poligono corrispondente.
Tali semiperimetri forniranno una successione di approssimazioni via via più precise, del numero π.
Fai in modo che sia l'utente a stabilire il numero di iterazioni.

ALCUNE FAMOSE E BELLE FORMULE PER IL CALCOLO DEL VALORE DI π

- $\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \ldots = \frac{\pi}{4}$ (Leibniz)
- $\frac{1}{1\cdot 3} + \frac{1}{5\cdot 7} + \frac{1}{9\cdot 11} + \frac{1}{13\cdot 15} + \frac{1}{17\cdot 19} + \ldots = \frac{\pi}{8}$
  (conseguenza della precedente)
- $\frac{2}{1}\cdot\frac{2}{3}\cdot\frac{4}{3}\cdot\frac{4}{5}\cdot\frac{6}{5}\cdot\frac{6}{7}\cdot\frac{8}{7}\cdot\frac{8}{9}\cdot\ldots = \frac{\pi}{2}$ (Wallis)
- $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \ldots = \frac{\pi^2}{6}$ (Eulero)
- $\frac{1}{1\cdot 2\cdot 3} - \frac{1}{2\cdot 3\cdot 5} + \frac{1}{3\cdot 4\cdot 7} - \frac{1}{4\cdot 5\cdot 9} + \ldots = \pi - 3$
  (Nilakantha)
- $\sqrt{12}\left(1 - \frac{1}{3\cdot 3} + \frac{1}{3^2\cdot 5} - \frac{1}{3^3\cdot 7} + \ldots\right) = \pi$ (Madhava)

In questa famiglia di formule, si hanno
- delle "somme infinite"
  (nel senso di: "somme di infiniti addendi")
- o dei "prodotti infiniti"
  (nel senso di: "prodotti di infiniti fattori")

e prendendo un certo numero di termini a partire da quello iniziale (es.: i primi 5, i primi 100 …) si ottiene un *valore approssimato* di π,
con l'approssimazione che diventa via via più precisa quanto più si fa alto il numero dei termini considerati.

Esercizio 40) - Vedi NOTA qui a fianco

Scrivi un programma Pascal che fornisca una approssimazione di π tramite una a tua scelta delle formule sopra riportate.
Fai in modo che sia l'utente a stabilire quanti termini utilizzare.

NOTA (il computer è COSTRETTO ad approssimare)

Un programma Pascal che calcoli il valore dei primi *n* termini in queste formule non potrà comunque, se *n* è molto alto, fornire il risultato *esatto* di quel calcolo, ma soltanto una sua *approssimazione*. La ragione è che una variabile numerica occupa in memoria un numero prefissato di byte (max 8-10) da cui tutta una serie di possibili errori di *overflow*, *underflow* (arrotondamento a 0 se il valore è troppo piccolo), "*cancellazione*".
C'è poi anche il fatto che il computer lavora in sistema binario, e nella conversione fra il decimale e il binario può essere costretto ad altre approssimazioni, dato il numero limitato di bit utilizzabili. Basti pensare che, ad esempio, il numero che in base dieci si scrive come 0.1 se viene portato in base due diventa periodico:

0.000110011001100110011001100...

Questi problemi sono inerenti alla natura stessa del computer, quindi vanno valutati anche da chi programmi in un linguaggio diverso dal Pascal.
Vedi ⇨ per approfondimenti.