

Esercitazione 08

2025/10/27 – Funzioni

L. Alessandrini

Credits: A. Montenegro per gli esercizi

Esercizio 00

Somma e Differenza

Testo

Scrivere un programma che chiede all'utente di inserire un numero naturale. Scrivere le funzioni:

- `leggiN()` : che legge da input e restituisce un numero naturale
- `somma()` : dati due interi, restituisce la somma dei due
- `diff()` : come la somma, ma per la differenza
- `sommaDiff()` : prende in input n_1 , n_2 , e restituisce sia $n_1 + n_2$ che $n_1 - n_2$

Nota! In generale quando si dice “prende in input” delle variabili, queste non sono TUTTE e SOLE le variabili in input, potrebbe esserci altro!

Esercizio 01

Divisori di un Numero

Testo

Scrivere un programma che chiede all'utente di inserire un numero naturale. Il programma stampa tutti i divisori del numero. Scrivere le funzioni:

- `leggiNumeroNaturale()` : che legge da input e restituisce un numero naturale
- `stampaDivisori()` : che dato un numero naturale, stampa tutti i suoi divisori

Esercizio 02

Numeri Amici

Testo

Scrivere un programma che:

1. prende in input due numeri naturali $n1$ e $n2$
2. verifica che i numeri siano amici
3. stampa se i numeri sono amici o meno

Scrivere le funzioni:

- `leggiNumeroNaturale()`: che legge e restituisce un numero naturale
- `sommaDivisori()`: che restituisce la somma di tutti i divisori di un numero dato in input
- `numeriAmici()`: che prende in input due numeri naturali e restituisce 1 quando i numeri sono amici, 0 altrimenti.

Nota: $n1$ e $n2$ sono amici se $n1$ è uguale alla somma dei divisori propri (i.e., eccetto se stesso) di $n2$ e viceversa.

Esercizio 02

Numeri Amici

Testo

Scrivere un programma che:

1. prende in input due numeri naturali $n1$ e $n2$
2. verifica che i numeri siano amici
3. stampa se i numeri sono amici o meno

Scrivere le funzioni:

- `leggiNumeroNaturale()`: che legge e restituisce un numero naturale
- `sommaDivisori()`: che restituisce la somma di tutti i divisori di un numero dato in input
- `numeriAmici()`: che prende in input due numeri naturali e restituisce 1 quando i numeri sono amici, 0 altrimenti.

Nota: $n1$ e $n2$ sono amici se $n1$ è uguale alla somma dei divisori propri (i.e., eccetto se stesso) di $n2$ e viceversa.

Nota! La funzione `sommaDivisori()` ritorna la somma di tutti i divisori ... Non solo i propri!

Esercizio 03

Numeri Perfetti

Testo

Scrivere un programma che chiede all'utente di inserire un numero naturale. Il programma verifica che il numero sia perfetto.

Nota: un numero si dice perfetto quando è uguale alla somma di tutti i suoi divisori (eccetto se stesso).

Scrivere le funzioni:

- `leggiNumeroNaturale()` : che legge da input e restituisce un numero naturale
- `sommaDivisori()` : che dato un numero naturale, restituisce la somma di tutti i suoi divisori
- `numeroPerfetto()` : che dato un numero naturale, restituisce 1 se il numero è perfetto, 0 altrimenti.

Esercizio 04

Numeri Perfetti, Abbondanti e Difettivi

Testo

Scrivere un programma che chiede all'utente di inserire un numero naturale. Il programma verifica che il numero sia perfetto, abbondante o difettivo.

Nota: un numero si dice perfetto quando è uguale alla somma di tutti i suoi divisori (eccetto se stesso). Il numero è abbondante se è minore della somma dei suoi divisori propri. Il numero è difettivo se è maggiore della somma dei suoi divisori propri.

Scrivere le funzioni:

- `leggiNumeroNaturale()` : che legge da input e restituisce un numero naturale
- `sommaDivisori()` : che dato un numero naturale, restituisce la somma di tutti i suoi divisori
- `numeroPerfettoPlus()` : che dato un numero naturale, restituisce 1 se il numero è perfetto, 0 altrimenti. Se il numero non è perfetto, la funzione deve essere in grado di comunicare al main se il numero è abbondante o difettivo.

Esercizio 05

Calcolo di Potenze

Testo

Scrivere un programma che calcola la potenza di un numero elevato a un esponente.

Il programma deve: chiedere due numero in input all'utente, uno sarà la base (anche negativo), l'altro l'esponente (da considerarsi solo positivo); calcolare la base elevata alla potenza.

Scrivere le funzioni:

- `leggiNum( )`: che legge e restituisce un numero intero
- `leggiNumPos( )`: che legge e restituisce un numero intero positivo
- `pow( )`: che prende in input base ed esponente e restituisce la base elevata all'esponente (base intera, esponente intero positivo)

Esercizio 06

Successione di Fibonacci

Testo

Scrivere un programma che:

1. chiede in input un numero naturale positivo
2. stampa la serie di fibonacci fino a quel numero

Scrivere le funzioni:

- `leggiNumeroPositivo()` :che legge e restituisce un intero positivo
- `stampaFibonacci()` : che prende in input un intero positivo e stampa la serie di fibonacci fino a qual punto

Esercizio 07

Calcolatrice 2.0

Testo

Scrivere un programma che, fin quando l'utente non ordina la terminazione inserendo 'q' o 'Q', permette di:

1. inserire due numeri
2. inserire un tipo di operazione: +, -, *, /
3. svolge e stampa a schermo il risultato dell'operazione

Scrivere le funzioni:

- `somma()`: presi due numeri ne restituisce la somma
- `differenza()`: presi due numeri ne restituisce la differenza
- `prodotto()`: presi due numeri ne restituisce il prodotto
- `divisione()`: presi due numeri ne restituisce la divisione e se il secondo operando è 0, comunica al chiamante che c'è stato un errore
- `leggiNumero()`: che legge e restituisce un numero lasciando il buffer pulito
- `leggiCarattere()`: che legge e restituisce un carattere lasciando il buffer pulito
- `menu()`: che implementa la logica della calcolatrice in un ciclo che dura fin quando l'utente lo vuole. Non ha nulla in input e non restituisce nulla

Thank You!

Mail: luca1.alessandrini@polimi.it

Website: <https://alessandriniluca.github.io/>