Dipartimento di Matematica e informatica
Ing. Cristiano Gregnanin
Laurea triennale in Matematica,
Esercizi di Programmazione

1. Acquisire due numeri da tastiera e stampare a video il maggiore
2. Acquisire dieci numeri da tastiera e stampare a video il quadrato di ciascuno di essi.
3. Acquisire da tastiera una sequenza di 10 numeri salvando ogni numero come elemento di una lista. Stampare il numero più grande presente nella lista.
4. Definire una funzione che, dati due numeri (che si assumono positivi) rappresentanti le lunghezze dei cateti di un triangolo rettangolo, restituisca la lunghezza dell'ipotenusa. Inserire quindi 2 numeri da tastiera e stampare la lunghezza dell'ipotenusa servendosi della funzione appena scritta.
5. Stampare a video tutti i numeri primi compresi fra 1 e 100.
6. Acquisire da tastiera 2 matrici di dimensione identica (2x3) e calcolarne la somma. Stampare quindi la matrice somma a video.
7. Usando le funzioni acquisire una sequenza di 10 numeri, inserirli in una lista e calcolarne il massimo. Stampare a video sia la lista che il massimo. Realizzare 3 funzioni: carica_lista, massimo_lista, stampa_lista
8. Scrivere un programma che prenda in ingresso da tastiera nome cognome e data di nascita di 5 persone. La struttura dati atta ad ospitare le informazioni della persona deve essere un dizionario.
   Esempio:
   {"nome":"mario","cognome":"rossi","datadinascita":{"year":1986,"month":12,"day":31}}. Si scriva una funzione per calcolare la persona più giovane e la persona più vecchia. Si stampi quindi a video il nome della persona più vecchia e quello della persona più giovane.

9. Acquisire una frase (che dovrà essere conclusa da un newline) da tastiera, e memorizzarla in un nuovo file di nome testo.txt. Al termine il file testo.txt dovrà quindi contenere la frase inserita in input dall'utente.
10. Acquisire una sequenza di frasi (ciascuna conclusa da un newline) da tastiera, e scrivere le frasi al termine del file *testo.txt* (ognuna in una riga diversa). Ad esempio se l'input è:

    frase 1
    frase 2
    frase 3

    Il contenuto del file testo.txt sarà:

    frase 1
    frase 2
    frase 3
11. dato un file testo.txt cosi struttato:

    questa è la prima frase dell esercizio
    questa è una frase ancora piu lunga
    questa è una frase corta

    leggere il contenuto del file e stampare a video ciascuna parola, in una riga distinta. Ad esempio:

    questa
    è
    la
    ….

    Suggerimento: si veda l'uso della funzione split()

12. Definire una funzione che, data una stringa e il nome di un file di testo, restituisca il numero di occorrenze di tale stringa all'interno del file. Per esempio, la sequenza *ci* compare due volte nel testo *Questo e un esercizio difficilissimo*, che invece non contiene occorrenze della sequenza *abc*.

    Firma della funzione:

    def ricerca(stringa,nomefile)

    Chiedere quindi in input da tastiera la stringa da ricercare e stampare a video il risultato della ricerca.
13. Dato il file di testo *esercizio13.txt*, definire una struttura dati per rappresentare le seguenti informazioni su un insieme di voli: *codice identificativo del volo* (sei

caratteri), *codice identificativo* (tre lettere) degli aeroporti di partenza e di arrivo, il giorno della settimana (lunedi, . . . , domenica) in cui il volo `è effettuato. Si legga quindi il file e si popoli una lista dove ogni elemento della lista è un dizionario che mappi la struttura delle righe del file. Definire quindi una funzione che riceva come argomento la lista e il codice di un aeroporto, e restituisca un dizionario contenente il numero di voli che partono da tale aeroporto per ogni giorno della settimana.
Esempio della struttura dati con cui mappare il file:
{“codiceVolo”:”xxxxxx”, “partenza”:”AAA”,”arrivo”:”BBB”,”giorno”:”lunedi”}
Esempio struttura dati di output:
{ "lunedi" : 0, "martedi" : 0, "mercoledi" : 0, "giovedi" : 0, "venerdi" : 0, "sabato" : 0, "domenica" : 0 }

14. Si assuma che un file di nome esercizio14.txt contenga gli esiti di un esame. Ogni riga contiene i seguenti dati su uno studente, separati da un carattere di spaziatura: numero di matricola, cognome (che si assume essere composto da una singola sequenza di caratteri) e voto (un intero compreso tra 18 e 30). Scrivere un programma che calcoli e stampi sullo schermo il voto medio conseguito dagli studenti. *Suggerimento*: si usi la funzione split()
15. Generare una lista con un numero di elementi elevato (>10000). Misurare il tempo impiegato per effettuare la ricerca di un numero all’interno della lista utilizzando i seguenti 4 algoritmi:
    a. Ricerca usando la funzione 'in' offerta da python
    b. Ricerca sequenziale scansionando tutta la lista alla ricerca dell’elemento.
    c. Ricerca binaria
    d. Ricerca binaria versione ricorsiva

    Suggerimento: per effettuare la misurazione usare la funzione *clock*() della libreria *time*. Per generare un numero casuale usare la funzione random() della libreria random. Per scegliere un numero casualmente ma che appartenga ad una lista usare la funzione choice(lista) della libreria random.
16. Si consideri un file di testo che contenga informazioni su un insieme di libri. Per ogni libro, in cinque righe separate sono indicati: il titolo, il codice ISBN, il nome dell'editore, l'anno di pubblicazione, il nome e il cognome degli autori (si assuma che ogni autore abbia un solo nome e un solo cognome). Le informazioni su diversi libri sono scritte in righe consecutive, senza righe vuote tra un libro e l'altro. Si veda come esempio il file allegato libri.txt

a. Definire un'opportuna struttura dati per memorizzare le informazioni su tutti i libri. Suggerimento: usare un dizionario
b. Scrivere tre funzioni che eseguano le seguenti operazioni:
    i. Dato il nome del file, acquisire i dati su tutti i libri, memorizzali in una variabile avente la struttura definita nel punto 1, e restituire il suo valore. *Suggerimento*: usare la funzione strip("\n") per rimuovere il carattere newline da una stringa. Esempio "stringa\n".strip("\n") produce "stringa". Osservare il risultato in console.
    ii. Dato il valore restituito dalla prima funzione, e il nome e cognome di un autore, stampare i titoli e l'anno di pubblicazione di tutti i suoi libri.
    iii. Dato il valore restituito dalla prima funzione, il nome di una casa editrice, e due anni (numeri interi), stampare il titolo e il codice ISBN di tutti i libri pubblicati da tale editore nell'intervallo di tempo tra i due anni (estremi inclusi)

    Scrivere le chiamate alle funzioni sopra definite, per memorizzare in una variabile di nome *libri* i dati contenuti nel file *libri.txt*; stampare il titolo e il codice ISBN dei libri pubblicati da *McGrawHill* tra il *2006* e il *2010;* stampare il titolo e l'anno di pubblicazione dei libri di cui sia autore o coautore Stefano Ceri.

17. Si implementino e si misurino i tempi di calcolo dei seguenti algoritmi di ricerca:
    a. insertion sort
    b. bubble sort
    c. selection sort
    d. quick sort
    e. metodo sort nativo di python. Esempio. lista.sort()

    Suggerimento: Usare una lista sufficientemente grande (1000). Per effettuare la misurazione usare la funzione *clock*() della libreria *time*. Per generare un numero casuale usare la funzione random() della libreria random.
18. Scrivere un programma che simula n volte il lancio di un dado e stampa le frequenze con cui ciascuna faccia del dado è uscita verificate a occhio se, per n sufficientemente grande, il dado è truccato esempio di output:
    "Dopo 10000 tiri, ecco il numero di volte che ogni faccia del dado è uscita:
    {'faccia2': 1695, 'faccia3': 1643, 'faccia4': 1700, 'faccia6': 1685, 'faccia5':1595, 'faccia1': 1682}"

Suggerimento: per generare un numero casuale compreso fra 1 e 6 usare: random.randint(1,6)

19. Creare un programma che inizializza una stringa g1 con valore casuale fra carta, sasso, forbice. Chieda all'utente di inserire un valore fra carta, sasso, forbice. Stampare il risultato del confronto secondo le regole del gioco carta sasso forbice. Si vogliono giocare 3 round. Se l'input dell'utente non e' corretto si passi direttamente al round seguente utilizzando il comando continue. Sono consentite in input sia stringhe maiuscole che stringhe minuscole. Suggerimento: usare la funzione lower() per normalizzare la stringa in input. Al termine dei 3 round fare un resoconto delle vittore, pareggi e sconfitte.
20. Scrivere una funzione sum_digits(n) che, dato un numero naturale n, restituisce la somma delle cifre in posizione pari e la somma delle cifre in posizione dispari di n
21. Manuela e Michele giocano a dadi. Scrivere una funzione che abbia come argomento il numero di manche e stampare su schermo chi vince ad ogni manche sapendo che il vincitore è chi totalizza il punteggio più alto tirando 4 dadi. La funzione dovrà stampare a schermo anche chi è il vincitore finale, cioè quello che vince più manche. Suggerimento: per tirare i dadi usare la funzione randint(min,max) della libreria random
22. Create una calcolatrice che le permetta di eseguire le seguenti operazioni: addizione, sottrazione, moltiplicazione, divisione, potenza (anche con base esponenziale) e la radice quadrata. (Definire una funzione per ogni operazione). Suggerimento: le funzioni che servono fanno parte tutte della libreria math e sono: exp(esponente) pow(base,esponente) sqrt(x)
23. Riprodure il "gioco dell'impiccato", scegliendo a caso la parola da indovinare tra quelle contenute in un file di nome "*parole_segrete.txt*" (le parole dovranno trovarsi in una o più righe, e dovranno essere separate solo da caratteri di spaziatura, come nel file di esempio). Suggerimento: usare la funzione choice della libreria random per scegliere una parola a caso fra quelle disponibili.
24. Dati due file, stampare le righe uguali (anche se si trovano in posizioni diverse). Nelle soluzioni i 2 file di esempio di chiamano *fileA.txt* e *fileB.txt*
25. Dato un file contenente del testo suddiviso in più righe, modificarlo cancellando tutte le righe che contengono una data parola scelta in input dall'utente. Nelle soluzioni il file di esempio di chiama *cancella-riga.txt*