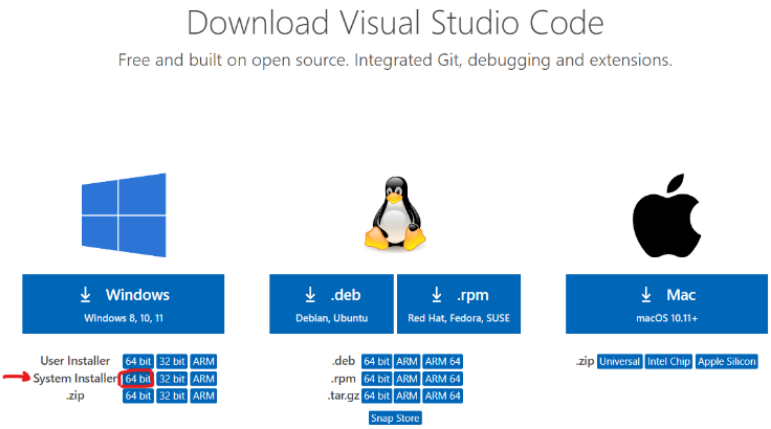
RACCOLTA DI MATERIALE ED ESERCIZI PER LA PROGRAMMAZIONE IN LINGUAGGIO PYTHON

Ambiente di sviluppo

1. Installazione dell'ambiente di lavoro:

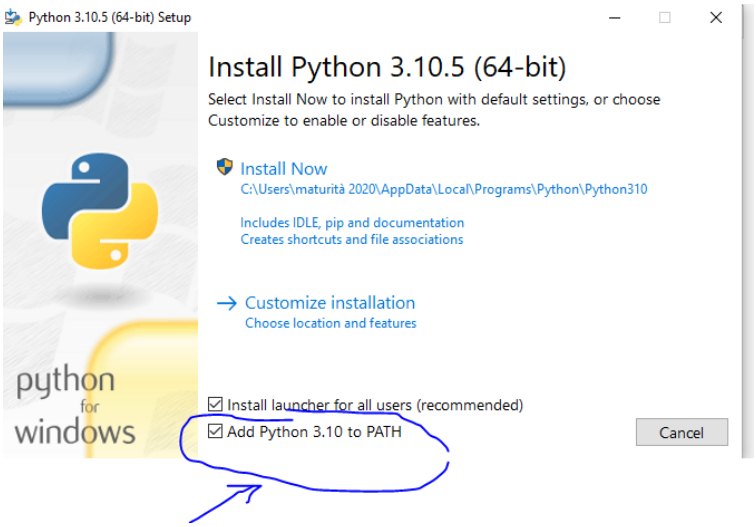
Installare Python che si può scaricare da:
https://www.python.org/downloads/



Consiglio, a meno di particolari necessità, di scaricare il system installer a 64bit.

Durante l'installazione:

- Flaggare l’opzione "Add Python x.xx to PATH" come mostrato nella seguente immagine;

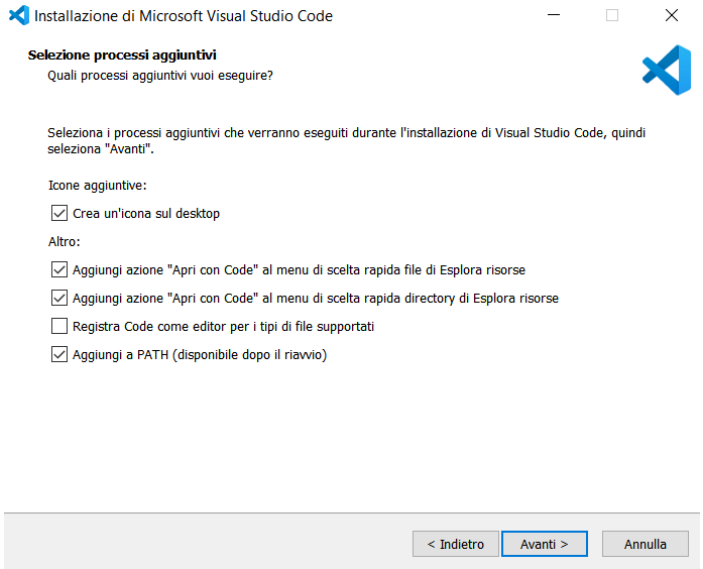


- Scegliere customize installation.
- Alla fase “optional features” lasciare tutto flaggato
- Alla fase “Advanced options” consiglierei, a meno di casi particolari, di aggiungere il flag alla prima voce, lasciare il flag agli altri 3 già flaggati e se si vuole flaggare anche la quinta voce.

Per verificare la corretta installazione basta eseguire in una qualsiasi console (cerca cmd su start) il comando “python” (python3 se su mac)

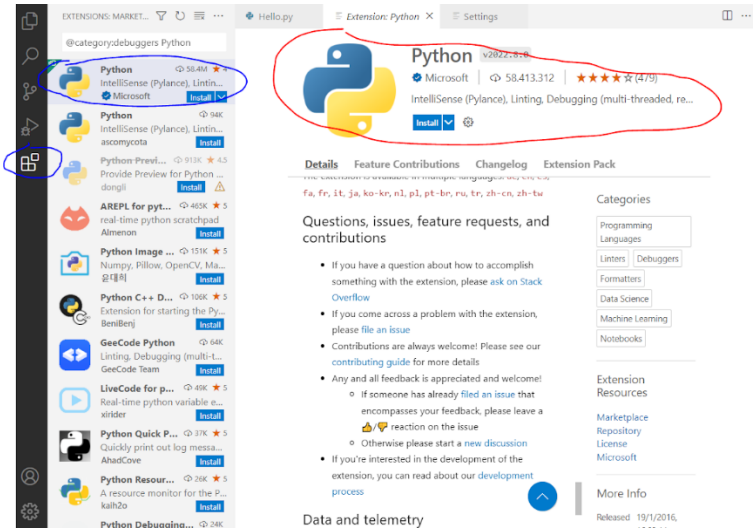
Per chi non lo avesse ancora fatto, installare visual studio code, consiglierei di usare, a meno di esigenze particolari, il system installer da 64bit che puoi trovare sul sito https://code.visualstudio.com/download .

Ricordarsi durante l’installazione di flaggare le voci corrette durante tutti i passaggi come mostrato dalla seguente immagine:

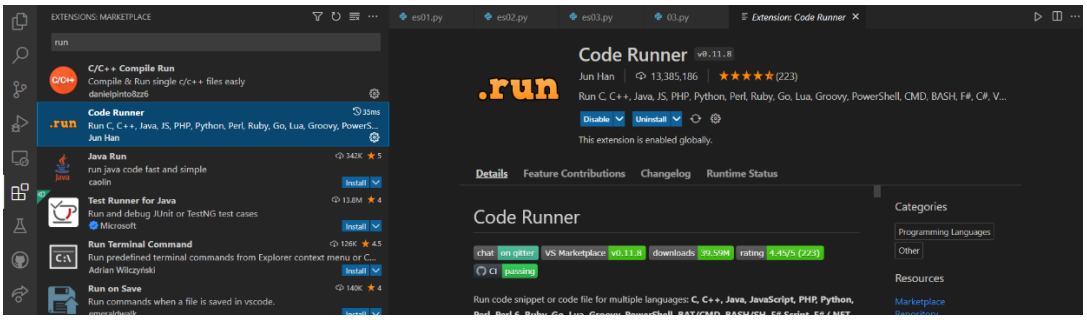


Per programmare comodamente in python con Visual studio code consiglio di installare le seguenti estensioni:

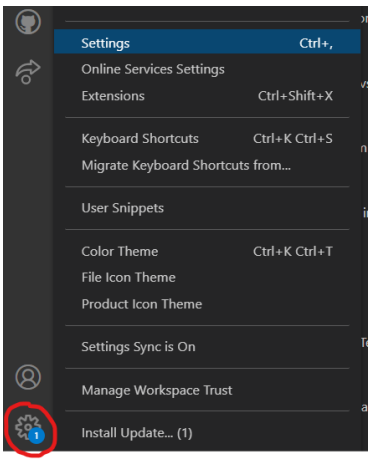
- Python.



- Code Runner

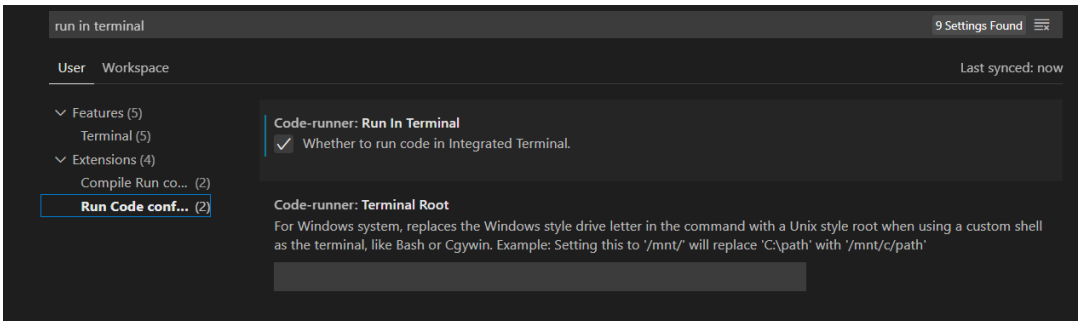


Una volta installate le estensioni vanno modificate alcune impostazioni. Il menù impostazioni è velocemente accessibile dal tasto in basso a sinistra:

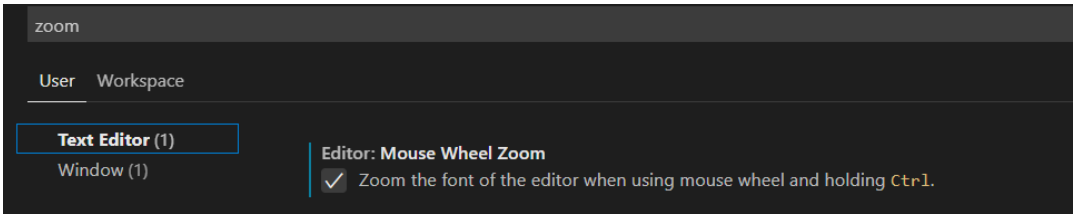


Le impostazioni da modificare sono:

- “run in terminal” da spuntare nelle impostazioni e trovabile velocemente inserendo la voce nella barra di ricerca e cliccando sull’estensione interessata come nella seguente immagine:



- "Mouse Wheel Zoom" da spuntare come fatto per l'impostazione precedente



È possibile modificare a piacere altre impostazioni come:

- Save file before run
- Save all files before run
- Color Theme
- Clear Previous Output

Indicazioni e materiale

Come manuale contenente esempi e spiegazioni di base su ogni aspetto del linguaggio consiglio di guardare il sito https://www.w3schools.com/python/ .
Online sono presenti molti altri tutorial anche in lingua italiana come ad esempio

1. https://www.programmareinpython.it/corsi-e-lezioni-python-dal-nostro-canale-youtube/
2. https://www.html.it/guide/guida-python/

Esercizi

La parte di eserciziario è stata inizialmente adattata dall'eserciziario per il linguaggio C, possono quindi essere presenti esercizi che fanno riferimento a elementi di tale linguaggio che devono ancora essere modificati.

Di seguito sono riportati gli esercizi normalmente svolti durante le lezioni del mio corso, voglio comunque riportare questo link (https://q2a.di.uniroma1.it/assets/eserciziario-python/it/script/) dove potete trovare un gran numero di esercizi, alcuni semplici altri anche piuttosto difficili, proposti per un corso universitario di informatica.

1. Variabili e operazioni elementari

1.1.1 Dati in input due numeri, scrivi il risultato di tutte le operazioni che è possibile fare con i due numeri (somma, sottrazione, moltiplicazione, divisione intera, resto della divisione, divisione con decimali, potenza)

Soluzione

1.1.2 Dato un numero decimale, ottieni e stampa:

1. Valore intero approssimato per difetto
2. Valore intero approssimato per eccesso
3. Valore intero approssimato in maniera intelligente
4. approssimazione del numero al secondo decimale
5. Valore assoluto del numero

Soluzione

1.1.3 Data l'area del cerchio, stampare la circonferenza

Soluzione

1.1.4 Data la base e l'altezza di un triangolo, scrivere l'area

Soluzione

1.1.5 Somma di due numeri (v1 - usando una terza variabile)

2. Algoritmi e strutture di controllo

2.1.1 Dati due numeri calcolare il loro quoziente se il divisore è != 0, ritornare "impossibile" se il divisore = 0

Soluzione

2.1.2 Dati in input due numeri, stampa il più grande tra i due

2.1.3 Dato in input un numero n, stampa la prima potenza di 2 maggiore o uguale a n

2.1.4 Moltiplicazione di due numeri senza usare l'operatore *

2.1.5 Divisione di due numeri interi senza usare l'operatore di divisione

Soluzione

2.1.6 dati dall'utente due numeri interi n1 e n2 il programma deve stampare:

2.1.7 tutti i numeri dal più piccolo dei due al più grande dei due

2.1.8 tutti i numeri dal più grande dei due al più piccolo dei due

2.1.9 tutti i numeri da n1 a n2 (comunque siano n1 e n2 quindi potresti dover andare in ordine crescente o decrescente)

2.1.10 tutti i numeri da n2 a n1 (comunque siano n1 e n2 quindi potresti dover andare in ordine crescente o decrescente)

2.1.11 Aggiungi all'esercizio sulla divisione tra due numeri senza usare l'operazione diviso anche la stampa del resto

2.1.12 Inserire n numeri != 0 (0 per finire), contare quanti sono i numeri inseriti

Soluzione

2.1.13 Variante avanzata: i numeri sono voti che devono essere validi e alla fine ne va calcolata la media.

2.1.14 Moltiplicazione di due numeri senza usare l'operatore di moltiplicazione

Soluzione

2.1.15 Dati tre numeri reali dire che tipo di triangolo essi formano (classificazione dei triangoli in base ai lati).

Soluzione

2.1.16 Data una sequenza di n numeri interi (valore di n dato dall'utente), calcolare la somma dei pari ed il prodotto dei dispari

Soluzione

2.1.17 Con un opportuno ciclo chiedere all'utente di inserire due numeri n1 e n2 compresi tra 1 e 100. Se i numeri non fossero corretti, rimanere nel ciclo e ripetere la richiesta. Stampare quanti sono i numeri pari compresi tra i due numeri

2.1.18 Data una sequenza di numeri terminata dal numero 0 (leggo numeri finchè non mi viene dato il numero 0), stampo il maggiore e il minore

2.1.19 Letto un numero intero positivo n stampare il fattoriale: $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$

2.1.20 Dati in input 3 numeri, stamparli in ordine crescente

2.1.21 Realizzare un programma che, presi in input 2 operandi reali e un operatore (+, -, *, /), esegue l'operazione stampandone il risultato

2.1.22 Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi terminati dal valore 0 (uno su ogni linea) e stampi il prodotto degli interi positivi e la somma dei negativi.

2.1.23 Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi fintanto che l'utente dice di volerne inserire ancora, e stampi il prodotto degli interi positivi e la somma dei negativi.

2.1.24 Dati due numeri, determinare il maggiore (verificare anche se sono uguali)

2.1.25 Data una sequenza di numeri terminata da 0, dire quanti sono i numeri inseriti (diversi da 0)

2.1.26 Dati due numeri in input b ed e, calcola e scrivi b^e in due modi diversi, usando l'operatore ** e senza poterlo usare.

2.1.27 Dati in ingresso 4 numeri, che rappresentano gli orari in cui avvengono due diversi eventi della giornata, in modo che i primi 2 numeri siano ore e minuti del primo orario e gli altri 2 numeri siano ore e minuti del secondo orario, stabilire quale evento è avvenuto prima. Inserisci dei controlli durante l'input in modo che le ore possano andare da 0 a 23 e i minuti da 0 a 59.

2.1.28 Dato un numero di minuti e un numero di secondi rappresentare il conto alla rovescia.

2.1.29 Dato in input un numero intero n, stampa l'ennesimo numero della successione di Fibonacci. La successione di Fibonacci è quella successione di numeri in cui ogni numero è la somma dei due numeri precedenti. I primi due numeri sono 1 1. La successione inizia così: 1 1 2 3 5 8 13 21 …

2.1.30 Dato in input un numero intero, conta da quante cifre è composto.

2.1.31 Dato in input un numero intero n di 3 cifre (in questo caso sarebbe utile mettere un controllo sull'input, cioè continuare a richiedere il numero fintanto che il numero dato non è di 3 cifre), stampa separatamente le sue cifre. Consiglio: usa divisioni per 10 e resti della divisione per 10. Puoi provare anche a generalizzare il programma in modo che funzioni anche con numeri di dimensione diversa da 3 cifre.

2.1.32 Generalizza l'esercizio precedente in modo che possa funzionare con un numero qualsiasi.

2.1.33 Dato in input un numero intero n, stabilisci se è primo

2.1.34 Dato in input un numero intero n, stampa la sua scomposizione in fattori primi

2.1.35 Data una sequenza di n numeri (n dato dall'utente) stabilire qual è il numero più grande, qual è il più piccolo e calcolare la media.

2.1.36 Data una sequenza di n numeri (n dato dall'utente) stabilire qual è il secondo numero più grande, qual è il secondo più piccolo.

2.1.37 Data una sequenza di numeri positivi (fai il controllo sull'input) terminati da 0, scrivi qual è la maggior differenza tra due numeri dati consecutivamente

2.1.38 Data una sequenza di numeri positivi (fai il controllo sull'input) terminati da 0, scrivi la somma dei numeri divisibili per 3

2.1.39 Dato un numero n scrivi somma e prodotto di tutti i numeri minori o uguali a n.

2.1.40 Dato un numero n stabilire quante volte è possibile dividerlo per 2. (esempio 20 è divisibile per 2, 2 volte)

2.1.41 Dato un numero n, decidere se è primo

2.1.42 Data una sequenza di prezzi di prodotti, calcolare la spesa totale sapendo che se un prodotto costa meno di 100€ lo devo scontare del 10% altrimenti del 5%. Decidi tu il metodo per capire quando terminare la lettura della sequenza di prezzi.

2.1.43 Data una sequenza di 5 numeri che rappresentano i voti presi nelle diverse materie, stabilire se lo studente sarà promosso, bocciato o rimandato a settembre. Lo studente è promosso se non ha insufficienze, è bocciato se ha almeno 3 insufficienze, altrimenti è rimandato. Ricordati di controllare i valori dei voti in input che devono essere voti validi.

2.1.44 Leggi 3 parole e stampale tutte insieme in un'unica volta (solo un parallelogramma di scrittura).

2.1.45 Dato un numero n positivo stampa tutti i numeri da 1 a n, i primi n multipli di 2 (2 compreso che consideri il primo multiplo) e i primi n multipli di 3

2.1.46 Dati tre numeri positivi verificare che questi tre numeri siano una terna pitagorica (una terna pitagorica è un insieme di 3 numeri per cui la somma del quadrato di due numeri sia uguale al quadrato del terzo numero, in altre parole sono le lunghezze dei lati di un triangolo rettangolo)

2.1.47 Scrivere l'algoritmo per il pagamento della spesa che consiste nel chiedere inizialmente se si è in possesso della carta fedeltà, poi chiedere tutti i prezzi dei prodotti acquistati terminando la sequenza con un prezzo uguale a zero. Il programma deve sommare i prezzi, e se si è in possesso della carta, scontare del 10% i prezzi minori di 50 e del 5% i prezzi maggiori di 50

2.1.48 Un libro deve essere restituito in biblioteca dopo 15 giorni di prestito altrimenti si è multati di 0,80€ al giorno di ritardo. Ricevuto in ingresso il numero di giorni di un prestito, visualizza se il socio deve essere multato per il ritardo e a quanto ammonta la multa da pagare.

2.1.49 Calcolare il costo della bolletta telefonica sapendo che i primi 30 scatti costano 20 centesimi l'uno, gli scatti dal 31 al 100 costano 15 centesimi l'uno, mentre gli ulteriori scatti costano 10 centesimi l'uno. Aggiungere infine una tassa fissa di 2,50€ per le spese telefoniche. In input al programma è dato il numero di scatti effettuati.

2.1.50 Costruire uno schema di flusso che rappresenti l'algoritmo per il seguente gioco: il computer genera un numero segreto (usa la funzione random(0, 100) che genera un numero casuale tra 0 e 99) e il giocatore deve individuarlo seguendo le indicazioni fornite dal computer (ti dice se il numero da trovare è più grande o più piccolo di quello che hai provato); una volta trovato il numero segreto, il numero di tentativi effettuati è visualizzato a video.

2.1.51 Calcolare il quoziente e il resto della divisione intera di due numeri interi positivi forniti in ingresso chiamati dividendo e divisore, applicando il metodo delle sottrazioni successive. Per esempio, se dividendo=13 e divisore=5, il programma dovrà restituire Quoziente=2 e Resto=3, calcolati sottraendo successivamente il valore di divisore dal valore di dividendo

2.1.52 Visualizza i termini della successione di Fibonacci compresi nell'intervallo tra due interi positivi N1 e N2, entrambi forniti in ingresso, con N2>N1 (controlla bene gli input)

2.1.53 Simula il lancio di un dado per N volte (con N intero e positivo in ingresso e usando la funzione random(7) per il lancio) e verifica che effettivamente la probabilità che esca 6 è 1/6. Per farlo fai tanti tiri e vedi se esce 6 un sesto delle volte (più tiri fai più dovresti avere un risultato positivo).

2.1.54 Simula il lancio di due dadi e verifica le probabilità che escano: una coppia di 6, un valore totale uguale a 7 (due diverse probabilità)

2.1.55 Dati in ingresso due numeri positivi x e y, visualizza in ordine decrescente la sequenza di numeri interi compresi tra x e y che sono divisibili per il minore tra x e y. Ad esempio, se x = 7 e y = 35, la sequenza è 35 28 21 14 7.

2.1.56 Simula una serie di lanci di un dado a 6 facce (d6) e un dado da 20 facce (d20). Calcola il valore medio che si ottiene lanciando il d6 e quello ottenuto lanciando il d20.

2.1.57 Valuta se è più probabile ottenere 7 oppure ottenere 8 lanciando 2 dadi da 6 e sommando i 2 valori ottenuti.

2.1.58 Voglio valutare la probabilità che tirando due dadi da 6 esca il valore 2. Il valore calcolato deve essere abbastanza preciso e decido di calcolarlo misurando la media di tanti tiri di dado e fermandomi solo quando vedo che facendo uscire un nuovo 2 valore calcolato non varia di più di 0,01%

2.1.59 Vengono dati in input i valori delle altezze di n persone (n chiesto all'utente). Per ogni persona oltre all'altezza viene indicato anche il sesso. Deve essere stampato il valore medio di altezza separatamente per i maschi e per le femmine. L'utente può decidere di non indicare né maschio né femmina, in quel caso non contare quella persona.

2.1.60 Scrivi il codice di un gioco per 2 persone basato sul lancio di dadi. All'inizio del gioco si decide che tipo di dado usare, si possono scegliere i dadi da 4, 6, 8 o 20 facce. Il gioco consiste nel tirare uno per volta il dado e sommando ogni volta i numeri usciti finchè uno dei due giocatori non supera un valore limite che è 4 volte il valore massimo rappresentato sul dado scelto (ad esempio se si sceglie il dado da 6 il limite è 24). Per rendere il gioco equo si può fare in modo che i due giocatori debbano tirare per forza lo stesso numero di tiri e che entrambi i giocatori possano superare il limite facendo finire il gioco in parità

2.1.61 Scrivi una variante del gioco precedente in cui l'obiettivo è avvicinarsi il più possibile al valore limite senza superarlo. Chi si avvicina di più vince. In ogni momento un giocatore può decidere di accontentarsi del valore ottenuto e smettere di tirare.

2.1.62 Leggi un numero positivo e controlla se esso è divisibile per 2 o divide 3 e non è compreso tra 10 e 20

2.1.63 Leggi un numero positivo e controlla se esso è divisibile per 2 o è un multiplo di 3 ed è compreso tra 10 e 20

2.1.64 dato un numero n stampa tutti i numeri da 1 a n che sono divisibili da 2 o 3 o 5 ma non sono divisibili per 4

2.1.65 Dato un numero positivo n disegna le figure geometriche riportate di seguito:

per n = 5

```
xxxxx        xxxxx        x
xxxx         xxxxx        xx
xxx          xxxxx        xxx
xx           xxxxx        xxxx
x            xxxxx        xxxxx
```

2.1.66 Esegui un programma che abbia il seguente output (chiaramente usando dei cicli):

```
1 2 3 4 5 6
2 3 4 5 6 +
3 4 5 6 + +
4 5 6 + + +
5 6 + + + +
6 + + + + +
```

2.1.67 Simula l'esecuzione di un gioco di dadi in cui due persone si scontrano lanciando ognuno un dado. Il gioco è diviso in round. Al primo round i due giocatori lanciano ognuno un dado da venti facce e vince chi fa il numero più alto, in caso di pareggio ritirano i dadi finchè uno dei due non vince. Chi perde il round deve affrontare i round successivi con un dado con una faccia in meno. Perde la partita chi arriva ad avere un dado con 0 facce. Mostra ad ogni round quante facce hanno i dadi dei due giocatori e mostra il risultato di ogni lancio dei dadi. Dichiara infine il vincitore.

2.1.68 Scrivi un programma che dato un numero n stampi una piramide come nel seguente esempio:

```
Con n = 3
  1
 121
12321
```

2.1.69 Scrivere un programma che legga due interi n e m con valori compresi tra 1 e 9, i cui prodotto sia inferiore a 35, e stampi m piramidi di altezza n. L'esempio si riferisce al caso n=3 e m=4.

```
  1    1    1    1
 121  121  121  121
12321123211232112321
```

2.1.70 Scrivi un programma che permetta di controllare i dati di input immessi dall'utente:

2.1.71 se l'utente inserisce un intero N compreso tra 1 e 10, il programma deve stampare a video il valore N^N,

2.1.72 se l'intero N e' compreso tra 11 e 20, il programma deve stampare a video la somma 1 + 2 + 3 +… + N

2.1.73 altrimenti deve dare un segnale di errore.

2.1.74 Si realizzi un programma che legga un intero N da tastiera, e stampi a video il risultato della seguente sommatoria:

$$\sum_{i=0}^{N} \left[(-1)^i \frac{4}{2 * i + 1} \right]$$

Una volta calcolato e stampato il valore a video, il programma deve chiedere un nuovo numero all'utente e ripetere il calcolo. Il programma deve terminare solo qualora l'utente inserisca un valore negativo.

2.1.75 Si progetti in C un programma che legge un float, rappresentante un ammontare di euro; di seguito il programma deve leggere un tasso d'interesse (in percentuale), ed un numero di anni. Il programma deve stampare, in uscita, per ogni anno, come l'ammontare cresce con gli interessi. Si ricordi che l'interesse si calcola con la seguente formula:

$C_{fin} = C_{in} (1 + (r/100))^N$

dove C_{fin} è il capitale finale, C_{in} è quello iniziale, r è l'interesse, e N rappresenta il numero di anni in cui si applicano gli interessi.

Ad esempio supponiamo che il capitale iniziale sia di 1000.0 €, con un tasso del 3%, per un periodo di 3 anni. L'output stampato deve avere all'incirca questo aspetto:

```
Capitale iniziale: 1000.00€
Dopo 1 anno: 1030.00 €
Dopo 2 anni: 1060.90 €
```

```
Dopo 3 anni: 1092.73 €
```

2.1.76 Realizzare un programma che legga da input un carattere dell'alfabeto e stampi a video il carattere stesso ed il suo valore ASCII. Il programma deve controllare che il carattere inserito sia compreso tra 'a' e 'z' o tra 'A' e 'Z' (in caso contrario si stampi un messaggio di errore). Dopo la stampa, il programma deve continuare a chiedere nuovi caratteri, finché l'utente non inserisce il carattere corrispondente al numero zero ('0'): in tal caso il programma termina.

2.1.77 Realizzare un programma che prende in input una sequenza di caratteri '0' e '1' e conta la lunghezza della più lunga sotto-sequenza di '0' di fila L'inserimento della sequenza termina quando si inserisce un carattere diverso da '0' e '1' A quel punto, si stampa a video il valore trovato

2.1.78 Realizzare un programma che prenda in input una sequenza di cifre (tra 1 e 9) e calcoli la somma massima fra le sottosequenze di cifre non decrescenti Il programma termina quando viene inserito lo 0

Esempio:

2	2	4	5	3	9	3	1	5	0
13				12		3	6		

2.1.79 Data in input una sequenza di n numeri (n dato dall'utente), stampare per ognuno il numero di quadrati perfetti minori di quel numero.

2.1.80 Conta quanti tiri di moneta devi fare prima di vedere uscire testa per 10 volte di fila.

2.1.81 Conta quanti tiri di un dado da 6 devi fare prima di vedere uscire lo stesso numero 5 volte di fila.

3. Liste, stringhe e matrici

3.1 Liste di soli numeri

3.1.1 Leggi n voti, caricali in un array, calcolane la media, poi stampa tutti i numeri e la media.

Soluzione

3.1.2 Continua l'esercizio precedente calcolando e stampando la media dei soli voti sufficienti.

3.1.3 Carica un array di numeri casuali, chiedi poi all'utente un numero e digli se questo numero è presente nell'array.

Soluzione

3.1.4 Carica un array, di dimensione a piacere, di numeri casuali compresi in un range di numeri deciso dall'utente. Esegui poi le seguenti operazioni:

1. inverti l'ordine dei numeri contenuti nell'array,

2. Date due posizioni a e b dell'array (chiaramente controlla che a sia minore di b e che siano compresi tra 0 e N-1) stampa prima i numeri dell'array dalla posizione a alla posizione b poi i numeri dalla posizione b alla posizione a
3. Dato un numero positivo x qualsiasi stampa tutti i numeri dell'array dalla posizione 0 alla posizione x. Se x > N-1, dopo aver stampato tutto l'array ricomincia a stampare dalla posizione 0 e continua fino ad aver stampato x+1 numeri.

3.1.5 Scrivi una funzione a cui vengono passati come parametro un elemento e una lista di elementi, e che ti dica in output se l'elemento passato sia presente o meno nella lista.

Qualora l'elemento sia presente nella lista, la funzione dovrà inoltre comunicarci l'indice dell'elemento.

3.1.6 Dato un array di numeri interi casuali, calcola e stampa il valore massimo e il valore minimo contenuti nell'array

3.1.7 Dato un array di numeri interi casuali, calcola e stampa media e mediana dell'array

3.1.8 Riempi un array di numeri casuali poi rappresenta l'array come un istogramma.

```
Ad es.
1: x
0:
4: xxxx
6: xxxxxx
3: xxx
9: xxxxxxxxx
```

3.1.9 Riempi un array di numeri casuali tra 0 e 9 e calcola la somma massima fra le sottosequenze di cifre non decrescenti. Il valore deve essere calcolato da un'apposita funzione.

Esempio:

2	2	4	5	3	9	3	1	5	0
13				12		3	6		

3.1.10 la prima riga rappresenta l'array e la seconda i vari valori calcolati (viene restituito 13 che è il più grande)

3.1.11 Scrivi un programma che riempia un array di numeri casuali di una cifra in modo tale che ogni numero sia sempre maggiore o minore di entrambi i numeri adiacenti (ad esempio 1 8 3 5 2 7 4 5 3). L'array generato deve essere stampato poi ordinato e infine ristampato. Per implementare le funzionalità descritte realizza due funzioni: una che riceve un array e lo riempie di numeri casuali come descritto, una che riceve un array e un indicazione che dica se ordinare l'array in ordine crescente o decrescente (ad esempio una variabile booleana).

3.1.12 Dato un array di numeri interi casuali, stampa tutti i numeri superiori alla media partendo dal fondo

Soluzione

3.1.13 Scrivere un paio di funzioni che calcolano perimetro e area di un triangolo rettangolo. Chiedere all'utente l'inserimento dei due cateti (i cateti vanno inseriti in una sola riga, in

modo da forzare l'utilizzo della funzione split()). Richiamare le funzioni e stampare i risultati.

Soluzione

3.1.14 Scrivi un programma che esegua le seguenti operazioni:

1. Chiedere con un ciclo all'utente di inserire i voti presi in matematica ed inserirli in una lista, uscire dal ciclo quando l'utente inserisce un valore negativo;
2. richiamare la funzione MediaVoti1() che restituisce la media dei voti presenti nella lista;
3. richiamare la funzione MediaVoti2() che restituisce la media di tutti i voti escludendo il più basso e il più alto. (si consiglia di usare il metodo sort() delle liste);

Variante: modificare l'esercizio in modo che i voti vengano inseriti in una sola riga con un solo comando input.

Soluzione

3.1.15 Scrivere una funzione che con un ciclo while chiede all'utente di inserire i 4 voti delle ultime verifiche di matematica (ed inserirli in una lista). Se il voto è inferiore a 1 o seperiore a 10, non va considerato. Solo quando l'utente ha inserito 4 voti corretti, uscire dal ciclo. Stampare la media dello studente in Matematica

3.1.16 Supponendo di avere una classe di 20 alunni (numerati nel registro da 1 a 20), il docente di matematica decide di dedicare i 5 appuntamenti di una settimana ad interrogare 4 studenti casuali, ogni giorno. Sapendo che ogni studente può essere interrogato al massimo due volte, stampare in ognuno dei 5 giorni i quattro interrogati, stampare quali studenti non sono mai stati interrogati.

3.1.17 Scrivere una funzione guessMyAge() che deve indovinare la mia età e stampare in quanti tentativi ha indovinato la mia età.

Nel programma principale chiedere all'utente di inserire la sua età e poi invoca la funzione guessMyAge() per indovinare il valore inserito.

3.1.18 Dichiarare due liste:

Con opportuni cicli while, riempire le due liste con 5 valori CRESCENTI casuali compresi tra 1 e 100

(naturalmente il primo valore non può superare 96, il secondo 97 ... l'ultimo 100)

Stampare il contenuto delle due liste

Stampare il quinto valore (in ordine di grandezza) considerando i dieci valori presenti nelle due liste

Soluzione

3.2 Stringhe

3.2.1 Scrivi una funzione che, dato un carattere in ingresso, restituisca in output il codice ASCII associato al carattere passato.

Soluzione

3.2.2 Stampa una stringa di una sola riga e una stringa formata da più righe

Soluzione

3.2.3 Scrivi una funzione (puoi farlo anche senza la funzione se non le hai ancora imparate) a cui viene passato un carattere come parametro, e risponde dicendo se il carattere è o meno una vocale.

3.2.4 Leggi una stringa da input e poi comunica all'utente di quante lettere è composta la stringa. Il conteggio deve essere fatto in due modi diversi, usando la funzione len e contando i caratteri uno alla volta.

Soluzione

3.2.5 Data la seguente stringa stampa:

stringa = " Vediamo come me la CAVO " # attento che ci sono spazi extra alla fine e all'inizio

1. La stringa così com'è
2. La stringa rimuovendo eventuali spazi iniziali e finali
3. La stringa nella versione con tutte le lettere minuscole
4. La stringa nella versione con tutte le lettere maiuscole
5. La stringa con la sola prima lettera maiuscola
6. La stringa con le sole prime lettere di ogni parola maiuscole
7. La seconda parola della stringa
8. La stringa ottenuta sostituendo tutte le 'a' con delle 'e'

Soluzione

3.2.6 Leggi una stringa da input e poi stampala in due modi diversi, la prima volta tutta insieme poi stampando una lettera per volta separando le lettere con uno spazio.

Soluzione

3.2.7 Leggi una stringa da input e poi stampala al contrario, cioè partendo dal fondo ma non modificandone il contenuto.

Soluzione

3.2.8 Leggi una stringa da input e poi modificala in modo da ottenere la stringa inversa (ad esempio “ciao” diventa “oaic”). Stampa infine la stringa.

Soluzione

3.2.9 Dichiara e inizializza due stringhe con valori a piacere, stampale. Scambia il contenuto delle due stringhe in modo che la prima contenga il contenuto della seconda e viceversa, poi ristampale.

3.2.10 Leggi da input due stringhe, se le due stringhe sono uguali comunica che sono uguali altrimenti scrivi quella che viene prima in ordine alfabetico

Soluzione

3.2.11 Leggi usando un’unica funzione input tre parole. Indica poi qual è la parola più lunga, la parola che viene prima in ordine alfabetico e la parola che contiene più vocali. (Questo esercizio è molto bello se si usa la funzione sort)

Soluzione

3.2.12 Data una lista di stringhe, crea e poi stampa un’unica stringa contenente tutte le stringhe della lista separate da “; “ (puntoevirgola e spazio)

Ad esempio dalla lista

["casa", "sedia", "cavallo", "andare in bici", "uscire di casa", "che bello!"]

viene creata la stringa

“casa; sedia; cavallo; andare in bici; uscire di casa; che bello!”

Soluzione

3.2.13 Scrivi e utilizza una funzione che riceve una stringa e un carattere e che stampi tutto il contenuto della stringa andando a capo ogni volta che trova il carattere dato (il carattere rappresenta un separatore della stringa in sottostringhe).

3.2.14 Scrivi un programma che contenga in un array la seguente stringa “Ciao a tutti, mi chiamo Francesco\nOggi vi voglio parlare delle stringhe, siete d’accordo?” e che la stampi in modo tale che quando trova una virgola va a capo(oltre a stampare la virgola) e quando trova un a capo, va a capo due volte.

3.2.15 Scrivi due funzioni che implementino il cifrario di Cesare cioè un cifrario che per cifrare un testo trasforma ogni lettera nella lettera che si trova tre posizioni più avanti nell’alfabeto. Le due funzioni sono le funzioni “cifra” e “decifra”. Scrivi infine u programma che permetta di inserire un testo da cifrare o decifrare a scelta.

Soluzione

3.2.16 In Svezia, i bambini giocano spesso utilizzando un linguaggio un po' particolare detto "rövarspråket", che significa "linguaggio dei furfanti": consiste nel raddoppiare ogni

consonante di una parola e inserire una "o" nel mezzo. Ad esempio la parola "mangiare" diventa "momanongogiarore".

Se sai usare le funzioni, scrivi una funzione in grado di tradurre una parola o frase passata tramite input in "rövarspråket" se no implementa il programma senza funzioni.

3.2.17 Scrivi una funzione generatrice di password.

La funzione deve generare una stringa alfanumerica di 8 caratteri qualora l'utente voglia una password semplice, o di 20 caratteri qualsiasi (anche simboli) qualora desideri una password più complicata.

3.2.18 Un indirizzo MAC (Media Access Control address) è un indirizzo univoco associato dal produttore, a un chipset per comunicazioni wireless (es WiFi o Bluetooth), composto da 6 coppie di cifre esadecimali separate da due punti.

Un esempio di MAC è 02:FF:A5:F2:55:12.

Scrivi una funzione genera_mac che generi degli indirizzi MAC pseudo casuali.

3.2.19 Scrivi e utilizza una funzione che riceve due stringhe e sia in grado di dire se la stringa più piccola è una sottostringa di quella più grande. Nel tuo programma il main deve creare due stringhe di lunghezza a piacere contenenti lettere casuali (solo lettere maiuscole e minuscole senza altri simboli o cifre; la funzione non cambia)

Variante: invece di dire se la stringa è contenuta nell'altra stringa restituire la posizione di partenza della sottostringa trovata.

Ad esempio "mi c" è sottostringa di "Ciao a tutti mi chiamo Francesco"

3.2.20 Scrivi un programma che chieda all'utente nome e cognome e poi:

1. verifichi che il nome e il cognome siano stati scritti con le iniziali maiuscole (devi controllare se la prima lettera ha i valori corretti secondo la tabella ASCII, i caratteri sono come numeri);
2. crei una stringa che contenga le generalità dell'utente (nome e cognome insieme);
3. stampi le generalità al contrario;
4. chieda nuovamente nome e cognome e verifichi se sono state riscritte le stesse cose di prima.

3.2.21 Data la seguente lista che contiene alcuni paesi europei

```
eu = ["italia", "germania", "svizzera", "francia", "spagna", "regnoUnito", "belgio" ]
```

e la seguente lista di stringhe (fatta da coppie città paese)

```
s = [
    "madrid spagna istanbul turchia bruxelles belgio ostenda belgio hannover germania berlino germania aleppo siria",
    "roma italia milano italia berna svizzera anversa belgio toledo spagna londra regnoUnito",
    "damasco siria  charleroi belgio munchen germania homs siria preston regnoUnito teheran iran",
    "bilbao spagna brema germania zurigo svizzera torino italia beirut libano marsiglia francia",
    "gerusalemme israele ankara turchia kobane siria colonia germania francoforte germania lione francia",
    "lugano svizzera siviglia spagna mosul iraq dortmund germania bordeaux francia manchester regnoUnito"
]
```

stampare i nomi dei due paesi europei, con il maggior numero di città nell'elenco

3.2.22 Le seguenti stringhe della lista, contengono il nome di una regione e alcune delle sue province:

```
ligureLombardo =
[
    "Lombardia Monza Milano Como Lecco Varese",
    "Liguria Genova",
    "Lombardia Pavia Lodi Cremona Mantova",
    "Liguria LaSpezia Savona Imperia",
    "Lombardia Brescia Bergamo Sondrio"
]
```

1. Riconoscere e memorizzare opportunamente le 12 province lombarde.
2. Stampare le 12 province in ordine alfabetico

3.2.23 Scoprire le città presenti in entrambi i tour di Ligabue e di Vasco

```
tourLigabue =
[
    "Prato 10 giugno",
    "Lucca 12 giugno",
    "Arezzo 15 giugno",
    "Siena 16 giugno",
    "Firenze 17-19 giugno",
    "Viareggio 20 giugno",
    "LaSpezia 22 giugno",
    "Genova 23-25 giugno",
    "Savona 27 giugno"
]

tourVasco =
[
    "Modena 3-5 giugno",
    "Parma 7 giugno",
    "Pavia 9 giugno",
    "Genova 10-11 giugno",
    "Carrara 14 giugno",
    "Lucca 16 giugno",
    "Pisa 17 giugno",
    "Livorno 18 giugno",
    "Firenze 20-23 giugno",
    "Arezzo 25 giugno",
    "Perugia 27 giugno"
]
```

3.2.24 La seguente lista descrivere i più celebri virologi Italiani nell'epoca covid

(Nome e Cognome - Ospedale - città - (O) Ottimista (P) Pessimista

```
covid= [
   "Roberto Burioni - San Raffaele - Milano (P)",
   "Matteo Bassetti - San Martino - Genova (O)",
   "Alberto Zangrillo - San Raffaele - Milano (O)",
   "Massimo Galli - Luigi Sacco - Milano (P)",
   "MariaRita Gismondo - Luigi Sacco - Milano (O)",
   "Andrea Crisanti - Università di Padova - Padova (P)",
   "Ilaria Capua - One Health - Florida (P)",
   "Antonella Viola - Città della Speranza - Padova (P)",
   "Fabrizio Pregliasco - Galeazzi - Milano (O)",
   "Walter Ricciardi - Policlino Gemelli - Roma (P)",
   "Franco Locatelli - Bambin Gesù - Roma (P)"
]
```

1. stampare nome e cognome dei virologi ordinati per cognome (si consiglia si usare la split() con il separatore "-";
2. stampare il numero di virologi ottimisti e di quelli pessimisti.

si ricorda che il metodo find() delle stringhe, restituisce -1 se l'elemento non è trovato.

3.2.25 Per ognuno dei seguenti numeri, stampare se è presente la cifra "tre", e nel caso quante volte è presente:

lista = [

"trentatre", "milletrecento","milequattrocentonovantadue", "trecentotredici"

]

3.2.26 Scrivi una semplice funzione rimario, a cui viene passato un elenco di parole come parametro e che riceva una parola inserita dall'utente tramite la funzione input.

La funzione rimario dovrà confrontare la parola inserita dall'utente con quelle presenti nell'elenco passato, alla ricerca di rime, intese come parole le cui ultime 3 lettere siano uguali alla parola inserita dall'utente.

Le rime dovranno essere quindi mostrate in output dall'utente.

3.3 Espressioni regolari

3.3.1 Per ognuno dei seguenti numeri, stampare se è presente la cifra "tre", e nel caso quante volte è presente:

lista = [

"trentatre", "milletrecento","milequattrocentonovantadue", "trecentotredici"

]

Soluzione

3.3.2 Scrivi una semplice funzione rimario, a cui viene passato un elenco di parole come parametro e che riceva una parola inserita dall'utente tramite la funzione input.

La funzione rimario dovrà confrontare la parola inserita dall'utente con quelle presenti nell'elenco passato, alla ricerca di rime, intese come parole le cui ultime 3 lettere siano uguali alla parola inserita dall'utente.

Le rime dovranno essere quindi mostrate in output dall'utente.

3.3.3

3.4 Liste miste

3.4.1 La seguente lista, contiene i nomi di una materia e i voti conseguiti

(il numero di voti, è diverso a seconda della materia)

lista = ["matematica", 5.5, 6, 4, "storia", 8,9, "italiano", 7, 7.5, 7, "inglese", 6.5 ,6, 5.5, "fisica", 6, 5.5]

1) stampare per ogni materia la media dei voti

Si consiglia di usare la funzione isinstance() per determinare se si tratta di una stringa o di un numero

3.4.2 la seguente lista di stringhe descrive una serie di operazioni

listaOp = ["15.5 + 13.8" ,

"16.4 / 2",

"7 * 2", "6.5 - 4.1"],

stamparne i risultati di queste operazioni (dove sono presenti due operandi separati da un operatore)

3.4.3 5 alunni (denominati alunno_1, alunno_2 alunno_3 alunno_4 alunno_5)

hanno ricevuto in storia rispettivamente 2, 3, 4, 3, 4 voti

e questa informazione è contenuta nella listaN:

listaN = [2, 3, 4, 3, 4]

la seconda lista riporta i sedici voti

listaVoti = [8.5, 7, 6, 4.5, 7, 5, 5.5, 7, 6.5, 9, 8.5, 7, 7, 6.5, 5, 6.5]

3.4.4 Stampare il nome dell'alunno che ha presto il voto più alto (alunno_3 con voto 9)

3.4.5 stampare i nomi delli alunni e la loro media in storia, ordinati in base alla media

3.4.6 Scrivi un programma che simuli il gioco del Black Jack.

Una mano del gioco è gestita da una funzione che devi chiamare manoBlackJack() che riceve come parametro una lista contenente la puntata fatta da ogni giocatore partecipante (implicitamente quindi anche il numero di giocatori) e restituisce la quantità di denaro da dare alla fine ad ogni giocatore (0 se perde, il doppio di quanto puntato se vince o lo stesso di prima se pareggia)

La funzione deve simulare la gestione di un mazzo di carte con una opportuna struttura dati. Per semplificare il gioco la funzione manoBlackJack() utilizza un nuovo mazzo mischiato ad ogni mano (ben lontano dalla realtà).

La funzione shuffle consente di mescolare il mazzo (restituisce un mazzo di 52 carte pieno e mischiato)

Sia ai giocatori che al banco, vengono assegnate inizialmente 2 carte pescate dal mazzo, entrambe scoperte per i giocatori, una coperta per il banco.

I giocatori giocano a turno e viene chiesto loro inserire la propria decisione che può essere solo di due tipi "vedo" o "continuo". Il singolo giocatore continua a pescare carte finchè perde (somma più di 21) o decide di fermarsi. Finito il turno dei giocatori il banco gioca automaticamente secondo le regole standard del gioco (pesca fintanto che raggiunge meno di 17, da 17 in su si ferma):

Il programma principale chiede se un giocatore si vuole aggiungere alla partita e con quale puntata, finchè si aggiungono giocatori continua a chiederlo (massimo 7 giocatori), poi chiama la funzione per giocare una mano e alla fine chiede se i giocatori vogliono rimanere (se hanno perso ne devono mettere altri) o vogliono ritirare soldi dal tavolo (tutti e se ne vanno, una parte e rimangono). Il programma poi chiede nuovamente se ci sono nuovi giocatori che vogliono partecipare (sempre massimo 7)

Il programma termina quanto tutti i giocatori lasciano il tavolo.

Varianti per completare il programma

1. Gestisci il mazzo in maniera realistica, queste sono le regole:

   “Il Blackjack classico (o francese) si gioca con sei mazzi di carte da poker (o francesi), per un totale di 312 carte. In mezzo ad esse vengono mescolate una carta di plastica e quando, durante il gioco, si arriva ad essa verranno rimescolate tutte le carte.”

   “Normalmente una taglia o sabot che dir si voglia è composta da sei mazzi di carte,all'inizio del gioco vengono tolte le prima 5 carte e si mette una carta nera che normalmente viene messa verso la fine della taglia e tiene fuori gioco un trenta quaranta carte(questo nasce al fine di dare filo da torcere a chi usa la conta),si mescolano le carte con l'uscita della carta nera. oggi però è sempre più in uso l'utilizzo delle mescolatrici,delle macchinette nelle quali il croupier mette le carte della mano precedente e che automaticamente vengono mescolate e rimesse in gioco,sono in uso presso i casino svizzeri e quello di montecarlo,altri non saprei,mentre sia a saint vincent che a san remo vengono ancora oggi mescolate a mano,ovviamente con le mescolatrice la conta diventa impossibile.

   Tutto questo ovviamente per quello che riguarda i casino reali.”

3.5 Liste di liste

3.5.1 Creare una matrice che contiene le tabelline dei numeri dall' 1 al 10. Alla fine stamparla.

Soluzione

3.5.2 Scrivi un programma che istanzia una matrice n x m di interi casuali (n e m scelti da te). Oltre a stampare l'intera matrice, chiedi all'utente quale riga o quale colonna stampare e stampala sullo schermo

Soluzione

3.5.3 Scrivere un programma che riempia due matrici A e B di dimensione n x m e calcoli la matrice C = A + B e la matrice D = A x B (in realtà è un finto prodotto di matrici). Stampa le matrici ottenute. Per il calcolo della somma calcola normalmente C[i][j] = A[i][j] + B[i][j] mentre per la moltiplicazione hai due possibilità:

1. Versione semplice se non riesci proprio a fare la versione corretta: C[i][j] = A[i][j] x B[i][j].
2. Versione complicata: vai a guardare su internet come si calcola e scrivi l'algoritmo per farlo.

Soluzione

3.5.4 Scrivi un programma che data una matrice n x m di interi, scambia le righe pari con quelle dispari
variante: le dimensioni della matrice vengono scelte dall'utente tra i valori massimi 10 x 20.

3.5.5 Scrivi un programma che data una matrice n x n di interi, con n scelto dall'utente, scambia le righe con le colonne

3.5.6 Scrivi un programma che riempia una matrice n x m di numeri interi a caso, n e m sono scelti dall'utente e il range dei numeri è da 0 a x scelto anch'esso dall'utente. Determina poi il numero più frequente generato nella matrice. Il riempimento delle matrici e la ricerca del numero più frequente devono essere svolti da due apposite funzioni, mentre l'interazione con l'utente dev'essere lasciata al main.

3.5.7 Scrivi un programma che riempia una matrice n x m di numeri interi a caso, n e m sono scelti dall'utente e il range dei numeri è da 0 a x scelto anch'esso dall'utente. Determina poi il la riga o la colonna con la somma dei numeri più grande. Il riempimento delle matrici e la ricerca della riga o della colonna devono essere svolti da apposite funzioni, mentre l'interazione con l'utente dev'essere lasciata al main.

3.5.8 Scrivi un programma che:

1. crei e riempia di valori casuali una matrice di dimensioni a tua scelta,
2. ne stampi il contenuto sullo schermo,

3. chieda all'utente un numero e cerchi se il numero è presente nella matrice
4. se il numero è presente si stampino le coordinate della posizione nella matrice

3.5.9 Scrivi un programma che istanzia una matrice n x m di interi casuali (n e m scelti da te). Oltre a stampare l'intera matrice, chiedi all'utente quale riga o quale colonna stampare e stampala sullo schermo

3.5.10 data una matrice n x m di numeri casuali (n e m scelti da te), stampa l'indice della riga con la massima somma dei numeri contenuti. Stampa anche l'indice della colonna con la massima somma dei numeri.

3.5.11 Scrivi un programma che per mezzo di un'apposita funzione riempia una matrice di numeri casuali. Stampa poi il numero di numeri pari contenuti in ogni riga della matrice (riga per riga separatamente) e il numero di numeri dispari contenuti in ogni colonna della matrice. Anche in questo caso definisci una o due funzioni per farlo (a scelta).

3.5.12 Data la seguente lista di liste

```
highways2 = [
    ["A1", "Milano-Napoli", ["Lombardia", "Emilia Romagna", "Toscana", "Umbria", "Lazio", "Campania"], 759],
    ["A2","Salerno-ReggioCalabria", ["Campania", "Basilicata", "Calabria"], 442],
    ["A3","Napoli-Salerno", ["Campania"], 52],
    ["A4","Torino-Trieste", ["Piemonte", "Lombardia", "Veneto", "Friuli"], 524],
    ["A6","Torino-Savona", ["Piemonte", "Liguria"], 124],
    ["A7","Milano-Genova", ["Lombardia", "Liguria"], 133],
    ["A8","Milano-Varese", ["Lombardia"], 43],
    ["A10","Savona-Ventimigla", ["Liguria"], 113],
    ["A11","Firenze-Pisa", ["Toscana"], 81],
    ["A12","Genova-Cecina", ["Liguria", "Toscana"], 210],
    ["A13","Bologna-Padova", ["Emilia Romagna", "Veneto"], 116],
    ["A14","Bologna-Taranto", ["Emilia Romagna", "Marche", "Abruzzo", "Molise", "Puglia"], 743],
    ["A15","Parma-LaSpezia", ["Emilia Romagna", "Liguria"], 108],
    ["A16","Napoli-Canosa", ["Campania", "Puglia"], 172],
    ["A19","Palermo-Catania", ["Sicilia"], 191],
    ["A21","Torino-Piacenza-Brescia", ["Piemonte", "Emilia Romagna", "Lombardia"], 238],
    ["A22","Brennero-Modena", ["Alto Adige", "Trentino", "Veneto", "Emilia Romagna"], 315],
    ["A23","Palmanova-Tarvisio", ["Friuli"], 119],
    ["A24","Roma-Teramo", ["Lazio", "Abruzzo"], 159],
    ["A25","Torano-Pescara", ["Lazio", "Abruzzo"], 115],
    ["A26","Genova-Gravellona", ["Liguria", "Piemonte"], 197]
]
```

1. stampare tutte le autostrade che attraversano la Lombardia
2. stampare l'autostrada che attraversa più regioni
3. stampare, tutte le autostrade, ordinate in base al numero di regioni attraversate, e a parità, in base alla loro lunghezza
4. stampare per ogni regione, il numero di autostrade che la attraversano

5. stampare per ogni regione tutte le autostrade che la attraversano
6. stampare per ogni regione tutte le autostrade che la attraversano, ordinandole in base al numero di regioni attraversate

Soluzione

3.6 Array di stringhe o matrici di caratteri

3.6.1 Scrivi un programma che chieda all'utente una serie di nomi che devono essere tutti memorizzati in un array di stringhe (e sarà quindi una matrice, ad esempio char nomi[n][m] con n numero dei nomi e m numero massimo di caratteri per stringa). Il numero di nomi da inserire deve essere chiesto all'utente. Quando viene inserito un nome bisogna controllare che inizi con una lettera e se questa è minuscola deve essere fatta diventare maiuscola. Alla fine stampa tutte le stringhe.

3.6.2 Continua l'esercizio precedente aggiungendo le seguenti funzionalità:

3.6.3 Cerca e stampa il nome più lungo

3.6.4 Cerca e stampa il nome che in ordine alfabetico è il primo

3.6.5 Chiedi all'utente un nome e digli se il nome è presente nell'elenco dei nomi (usa strcmp).

3.6.6 Dato un array di stringhe (quindi una matrice), riempito come preferisci, stampa la stringa più lunga, la più corta e la lunghezza media delle stringhe. I tre risultati devono essere ottenuti per mezzo di apposite funzioni.

3.6.7 Scrivi e usa una funzione che dato un array di stringhe (quindi una matrice), restituisca il numero di parole palindrome contenute nell'array. Riempi l'array come preferisci.

3.6.8 Scrivi un programma che permetta di rappresentare il campo di gioco di "Campo minato". Il campo deve essere formato da una superficie suddivisa in caselle quadrate, quindi avrai una grande tabella (ad esempio di dimensioni 8x12 ma puoi scegliere diversamente). Sul campo devono essere disposte a caso una serie di mine (suggerisco 16 ma puoi cambiare). Fai poi in modo che ogni casella che non sia una mina abbia un numero che rappresenti il numero di mine adiacenti (anche in diagonale quindi un numero da 1 a 8) lasciando vuote le celle che non hanno mine adiacenti (non scrivere 0)

Soluzione

3.6.9 Crea un programma di gestione di una agenda. L'agenda deve permettere di visualizzare e inserire i dati riguardanti i propri contatti. Per ogni persona devono essere memorizzati nome cognome e numero di telefono. La visualizzazione dei dati deve avvenire in ordine alfabetico per nome o cognome (scegli tu). Aggiungi poi una funzione per cancellare un contatto in una specifica posizione dell'agenda. (Non avendo ancora fatto la gestione dei files ogni volta l'agenda partirà da uno stato predefinito che puoi decidere tu). Aggiungi anche un'interfaccia utilizzabile.

3.6.10 Scrivi un programma che chieda all'utente di scrivere una frase e le legga tutte in un unico input. Le parole della frase devono poi essere stampate una per volta andando sempre a capo tra una parola e l'altra.

3.6.11 Scrivi un programma che chieda all'utente di scrivere una serie di parole lette una ad una. Queste parole devono essere unite a formare un'unica stringa che deve essere infine stampata.

3.6.12 Scrivi un programma che chieda all'utente di scrivere una frase e le legga con la funzione gets o fgets (otterrai quindi un'unica stringa). Le parole della frase devono poi essere ordinate e stampate in ordine alfabetico. (consiglio: prendi le parole dalla stringa, mettile in un array e ordina l'array)

3.7 Algoritmi di ordinamento base

3.7.1 Carica un array di numeri casuali e utilizza il selection sort per ordinarlo. Deve essere fatta una stampa dell'array prima e dopo l'ordinamento

3.7.2 Carica un array di numeri casuali e utilizza il bubble sort per ordinarlo. Deve essere fatta una stampa dell'array prima e dopo l'ordinamento

3.7.3 Carica un array di numeri casuali e utilizza la funzione sort della libreria standard per ordinarlo. Deve essere fatta una stampa dell'array prima e dopo l'ordinamento

3.7.4 Scrivi un programma che dato un array di numeri interi, sia in grado di:

1. controllare se l'array è palindromo,
2. invertire l'ordine dei numeri dell'array,
3. ordinare in ordine decrescente l'array.

Soluzione

3.7.5 Scrivi un programma che ordini un array con l'algoritmo selection sort, prima in ordine crescente poi in ordine decrescente.

3.7.6 Scrivi un programma che ordini un array con l'algoritmo bubble sort, prima in ordine crescente poi in ordine decrescente.

3.7.7 Scrivi un programma che dato un array di dimensione n e due valori x y tali che x<=y<=n, ordini l'array dalla posizione x alla posizione y comprese.

3.7.8 Dato un array di stringhe, che puoi riempire come preferisci, ordina l'array in ordine alfabetico prima crescente e poi decrescente. Devono essere inserite le stampe dell'intero array prima e dopo ogni ordinamento.

Soluzione

3.7.9 Crea un programma di gestione di una agenda. L'agenda deve permettere di visualizzare e inserire i dati riguardanti i propri contatti. Per ogni persona devono essere memorizzati nome cognome e numero di telefono. La visualizzazione dei dati deve avvenire in ordine alfabetico per nome o cognome (scegli tu). (Non avendo ancora fatto la gestione dei files ogni volta l'agenda partirà da uno stato predefinito che puoi decidere tu).

3.7.10 Con la tecnica del Bubble Sort (2 cicli FOR e una IF):

1. ordinare la seguente lista di numeri in ordine crescente
   lista1 = [8, 5, 7, 12, 18, 17, 3, 5]

2. ordinare la seguente lista di capitali in ordine alfabetico
   lista2 = ["Tallinn", "Vilnius", "Riga", "Copenaghen", "Stoccolma", "Helsinki"]
3. ordinare la lista delle capitali in base alla lunghezza del nome della Capitale (e a parità di lunghezza in base al nome della capitale)

3.7.11 La seguente lista di stringhe contiene una serie di stringhe che descrivono alcuni celebri immunologi:

imLista = [

"Pasteur 1822 1895 Francia", "Koch 1843 1910 Germania", "Fleming 1881 1955 RegnoUnito",

"Dulbecco 1914 2012 Italia", "Montagnier 1932 2022 Francia", "Gallo 1937 1922 Usa",

"Gram 1853 1938 Danimarca", "Jenner 1749 1823 RegnoUnito", "Sabin 1906 1993 Usa",

]

1. stampare gli immunologi ordinati in base a quanti anni sono vissuti
2. stampare gli immunologi ordinati in base alla nazione, e a parità di nazione in base al nome

Soluzione

3.7.12 Data la seguente lista:

```
highways = [
    "A1 Milano-Napoli 759",
    "A2 Salerno-ReggioCalabria 442",
    "A3 Napoli-Salerno 52",
    "A4 Torino-Trieste 524",
    "A6 Torino-Savona 124",
    "A7 Milano-Genova 133",
    "A8 Milano-Varese 43",
    "A10 Savona-Ventimigla 113",
    "A11 Firenze-Pisa 81",
    "A12 Genova-Cecina 210",
    "A13 Bologna-Padova 116",
    "A14 Bologna-Taranto 743",
    "A15 Parma-LaSpezia 108",
    "A16 Napoli-Canosa 172",
    "A19 Palermo-Catania 191",
    "A21 Torino-Piacenza-Brescia 238",
    "A22 Brennero-Modena 315",
    "A23 Palmanova-Tarvisio 119",
    "A24 Roma-Teramo 159",
    "A25 Torano-Pescara 115",
```

"A26 Genova-Gravellona 197"

]

1. Stampare l'autostrada più corta e quella più lunga (Risultato A1 Milano-Napoli 759 A8 Milano-Varese 43)
2. stampare la classifica delle autostrade ordinate per lunghezza
3. stampare le prime tre città che sono capolinea di una autostrada

3.7.13 La seguente lista, contiene alternati i nomi di una materia e il voto finale nella materia

lista = ["matematica", 5.5, "storia", 8, "italiano", 7, "inglese", 6.5 , "fisica", 6]

1. stampare la materia che ha preso il voto più alto
2. utilizzando la tecnica del bubble sort - stampare le materie ordinate in base al voto
3. stampare le materie ordinate in base al voto

4. Funzioni base.

4.1.1 Scrivi e utilizza una funzione per calcolare e restituire la somma tra due numeri

4.1.2 Scrivi una funzione che prende due numeri come parametro e manda in print il più grande tra i due.

4.1.3 Scrivi e usa una funzione che, passata come parametro una lista di interi, restituisce il maggiore tra i numeri contenuti nella lista.

4.1.4 Scrivi e utilizza 4 funzioni che effettuano le 4 operazioni e restituiscono il risultato.

4.1.5 Scrivi e utilizza una funzione che dati due valori e un simbolo di una delle 4 operazioni effettua quella operazione e restituisce il risultato

Soluzione

4.1.6 Quando possibile, scrivi e utilizza le seguenti funzioni:

1. raddoppia un numero passando parametri per valore,
2. raddoppia un numero passando parametri per indirizzo,
3. raddoppia due numeri contemporaneamente.

Soluzione

4.1.7 Scrivi e utilizza una funzione che restituisca il prodotto di due numeri e alla fine dell'esecuzione lasci modificati i due numeri in modo che valgano il doppio di prima.

4.1.8 Funzione che date le coordinate di due punti nel piano, calcolino e restituiscano le coordinate del punto medio tra i due (senza approssimazioni).

Soluzione

4.1.9 Funzione che dato il tempo di caduta di un oggetto, calcola l'altezza da cui è caduto l'oggetto e la velocità di impatto. Le formule che ti servono sono: h = 1/2 * g * t^2, con g=9,81 e v=g*t.

Soluzione

4.1.10 Funzione che prevede se un oggetto si rompe cadendo da una certa altezza. Quello che ti serve sapere è quindi l'altezza da cui cade l'oggetto e la velocità massima di impatto che l'oggetto può sopportare. Le formule che ti servono sono: h = 1/2 * g * t^2, con g=9,81 e v=g*t.

Soluzione

4.1.11 Scrivi una funzione che data in ingresso una lista A contenente n parole, restituisca in output una lista B di interi che rappresentano la lunghezza delle parole contenute in A.

4.1.12 Scrivi ed utilizza una funzione che dati due numeri li aumenta entrambi di un certo valore passato come parametro.

4.1.13 Scrivi e utilizza una funzione che dato un array di numeri, trova e restituisce il massimo e il minimo numero presenti.

4.1.14 Scrivi e utilizza una funzione che, data una stringa, restituisce true se la stringa inizia con una lettera maiuscola.

4.1.15 Funzione che data una stringa, la stampa separando ogni carattere della stringa con uno spazio.

4.1.16 Scrivi e utilizza una funzione che riceve come parametri due stringhe e restituisce una stringa che è la concatenazione delle altre due.

4.1.17 Scrivi un programma che, dati n voti in un array, li passi ad una funzione che restituisca la media dei voti escludendo il voto più alto e il voto più basso.

4.1.18 Scrivi e utilizza una funzione che stampa una stringa ricevuta come parametro.

4.1.19 Scrivi una funzione che scambia il contenuto di due variabili intere (i valori devono rimanere scambiati per chi chiama la funzione).

4.1.20 Scrivi un programma che sappia riordinare un array. Per farlo dividi le seguenti funzionalità in diverse funzioni:

1. funzione che riceve un array e lo riempie di numeri casuali
2. funzione che stampa un array
3. funzione che riceve un array e lo ordina; puoi usare l'algoritmo che preferisci
4. main che dichiara un array e poi usando le funzioni descritte sopra lo riempie di numeri casuali, lo stampa, lo ordina e poi lo ristampa.

Soluzione

4.1.21 Scrivi e utilizza una funzione che dato un array restituisce i valori massimi e minimi dell'array

4.1.22 Scrivi e utilizza una funzione che calcola il quoziente tra due numeri interi e restituisce, oltre al quoziente anche il resto della divisione.

Soluzione

4.1.23 Scrivi e utilizza una funzione che data una stringa, fa in modo che la prima lettera sia maiuscola e tutte le altre minuscole. i caratteri che non sono lettere non devono essere modificati.

4.1.24 Scrivi una funzione che calcola il massimo comune divisore tra due numeri.
Scrivi ed utilizza poi una funzione che dati due valori numeratore e denominatore, semplifica la frazione, usando la funzione mcd appena definita.

4.1.25 Scrivi ed utilizza una funzione che riceva come parametri tre numeri e che restituisca true se questi numeri possono essere le misure di un triangolo. In questo caso la funzione deve anche restituire una stringa in cui c'è scritto il tipo di triangolo (equilatero, isoscele, scaleno).

5. Funzioni avanzate

5.1 Funzioni ricorsive

5.1.1 Scrivi ed utilizza una funzione iterativa che calcola il fattoriale di un numero

5.1.2 Scrivi ed utilizza una funzione ricorsiva che calcola il fattoriale di un numero

5.1.3 Scrivi e utilizza le versioni iterativa e ricorsiva di una funzione che dato un array di interi, calcola e restituisce la massima differenza tra due numeri consecutivi.

5.1.4 Scrivi una funzione iterativa che calcola l'ennesimo numero della successione di Fibonacci.

5.1.5 Scrivi una funzione ricorsiva che calcola l'ennesimo numero della successione di Fibonacci.

5.1.6 Scrivi una funzione ricorsiva che effettui la moltiplicazione tra due numeri

5.1.7 Scrivi una funzione che calcola ricorsivamente il quoziente tra due numeri utilizzando le sottrazioni.

5.1.8 Scrivi una funzione ricorsiva che stampi un array di numeri.

5.1.9 Scrivi una funzione ricorsiva che stampi al contrario un array di numeri.

5.1.10 Scrivi e utilizza le versioni iterativa e ricorsiva di una funzione che riceve un array e inverte l'ordine dei numeri in esso contenuti.

5.1.11 Funzione ricorsiva per trovare i valori massimi e minimi di un vettore.

5.1.12 Funzione ricorsiva per cercare un valore all'interno di un vettore non ordinato (ricerca lineare). Risultato: –1 se non lo trova, altrimenti l'indice della posizione dove è stato trovato.

5.1.13 Scrivere una funzione ricorsiva per cercare un valore all'interno di un vettore ordinato (ricerca dicotomica). Risultato: – 1 o l'indice.
Indicazioni: se il vettore è ordinato, iniziate a controllare il numero a metà del vettore, se il numero controllato è più piccolo di quello da trovare, cercherò nella parte del vettore a

sinistra (richiamo la funzione indicando come intervallo in cui cercare solo la parte a sinistra), se è più piccolo dovrò cercare a destra, se è uguale l'ho trovato.

5.1.14 Scrivere una funzione ricorsiva che riceve un array di interi e restituisce true se gli elementi sono tutti dispari e false se non lo sono.

5.1.15 Scrivere una funzione ricorsiva che riceve un array di interi e restituisce il prodotto degli elementi negativi.

5.1.16 Scrivi una funzione di ordinamento ricorsiva.

5.1.17 Si definisca la funzione ricorsiva (o che usa una vostra funzione ricorsiva) che presa in input una stringa di caratteri restituisce la lista delle diverse "sottoparole crescenti" di tale stringa. Le sottoparole devono comparire nella lista in ordine lessicografico (alfabetico).

Si ricorda che una sottoparola e' quello che si ottiene da una parola concellandone 0 o piu' caratteri (in testa, in coda o nel mezzo). Inoltre una sottoparola si dice crescente se i caratteri che la compongono letti da sinistra a destra risultano ordinati lessicograficamente.

Ad esempio con l'input "zanzara" si ottiene:

['a', 'aa', 'aaa', 'aar', 'an', 'anr', 'anz', 'ar', 'az', 'n', 'nr', 'nz', 'r', 'z', 'zz']

Soluzione

5.2 Algoritmi di ordinamento avanzati

5.2.1 prova

6. Classi

6.1.1 Scrivi un programma che definisca un punto e poi dati due punti scelti da te sia in grado, per mezzo di due apposite funzioni, di calcolare la distanza fra due punti e trovare il punto medio.

6.1.2 Definisci una struct punto. Scrivi ed utilizza poi le funzioni:

1. Funzione per stampare una rappresentazione del punto (es. A(1,2))
2. Funzione per calcolare il punto medio
3. Funzione per calcolare la retta passante tra due punti. Dovrai anche definire la struct retta che contiene le informazioni necessarie ad identificare una retta e definire una funzione per stamparla in modo standard (y=mx+c)
4. Funzione che verifica se tre punti sono allineati.

6.1.3 Scrivi un programma per gestire un corso. Al corso sono iscritte delle persone, massimo 30. Per ogni persona dovete sapere nome e cognome e ad ognuno alla fine del corso viene assegnato un voto da 1 a 10. Alla fine stampa tutti i dati di ogni studente, compresi i voti.

6.1.4 Completa il seguente programma in cui è stato dichiarato un array di stringhe in cui sono scritti nomi, cognomi e date di nascita e morte di esploratori famosi. Devi definire le strutture dati adeguate a contenere queste informazioni: una struct esploratore e un array di esploratori. Usa la funzione sscanf per leggere i dati dalla stringa dati e riempi le strutture dati da te definite. scrivi poi una serie di funzioni che lavorino sulle tue strutture dati:

1. una funzione che stampa l'array di esploratori.
2. una funzione che ordina l'array in ordine alfabetico secondo il nome degli esploratori
3. una funzione che ordina l'array in ordine crescente secondo le date di nascita degli esploratori

Consigli e indicazioni extra:

4. La funzione sscanf l'abbiamo usata pochissimo in passato ma trovate facilmente online indicazioni su come si usa.
5. Alcuni potrebbero trovare comodo definire una struttura dati contenente l'array degli esploratori insieme al numero di esploratori contenuti nell'array.

```
int main(){
   char dati[15][50] = {
       "Marco Polo 1254 1324",
       "Cristoforo Colombo 1451 1506",
       "Amerigo Vespucci 1454 1512",
       "Francisco Pizarro 1475 1541",
       "Ferdinando Magellano 1480 1521",
       "Hernan Cortez 1485 1547",
       "Walter Raleigh 1552 1618",
       "Henry Hudson 1570 1611",
       "James Cook 1728 1779",
       "Charles Darwin 1809 1882",
       "Kit Carson 1809 1866",
       "David Livingstone 1813 1873",
       "Charles Foucauld 1858 1916",
       "Ronald Amundsen 1872 1928",
       "Ernest Shakleton 1874 1922",
   };
}
```

6.1.5 Scrivi e utilizza un insieme di funzioni da utilizzare per fare calcoli con le frazioni. In pratica devono essere definite le funzioni per fare almeno le 4 operazioni (magari anche altre operazioni). Deve anche essere fatta una funzione per effettuare la semplificazione delle frazioni (puoi decidere se creare e usare anche un'altra funzione mcd o utilizzare un altro metodo). Scrivi tutte queste funzioni in un file separato e poi utilizza queste funzioni per fare dei calcoli con le frazioni in un programma.
In una prima versione del programma considera numeratore e denominatore come due variabili intere separate. In una seconda versione definisci invece una struct frazione che contiene numeratore e denominatore. Le funzioni dovranno essere modificate di conseguenza.

6.1.6 Scrivi un programma in cui è definita una struttura studente. Dello studente ci interessano il nome, i vori presi e la media dei voti. Il programma deve istanziare uno o più studenti e aggiungere una serie di voti casuali (anche mezzi voti ma non tutti i decimali possibili). Voglio creare uno studente e poi usare delle funzioni per aggiungere voti e calcolare la media. Infine stampare lo studente con tutti i suoi dati con una funzione apposita.

6.1.7 Definire e utilizzare tutte le funzionalità delle la classe TriangoloRettangolo di seguito descritta:

1. Il costruttore riceve le misure dei due cateti e autonomamente calcola anche la misura dell'ipotenusa
2. definire gli ulteriori metodi:
   1 Area(self)
   2 Perimetro(self)
   3 Stampa(self)

Soluzione

6.1.8 Definite le classi:

1. TriangoloRettangolo
2. Rettangolo
3. Quadrato
4. Rombo

Svolgi le seguenti operazioni:

5. dichiara una serie di oggetti di queste classi e inseriscile in una lista,
6. ordina la lista in base all'Area(),
7. invoca il metodo Stampa() su ogni oggetto della lista.

Soluzione

6.1.9 data la seguente lista di stringhe

```
listaTreni = [
    "Milano-Centrale Bologna  7:30  8:34",
    "Milano-Centrale Genova-Principe  8:05  9:44",
    "Milano-Centrale Torino  7:30  8:30",
    "Milano-Centrale Piacenza  8:01  8:53",
    "Milano-Centrale Chiasso  8:10  9:10",
    "Milano-Centrale Brescia  8:15  8:51","Milano-Centrale Verona  8:45  9:58","Milano-Centrale Varese  8:05  9:00",
    "Brescia Venezia 8:13  9:52",
    "Voghera Torino 7:57  9:25"
]
```

Definire una classe ViaggioTreno che riceve nella __init__ una stringa di listaTreni.

Gli attributi e i metodi sono scelti a piacere, con la sola condizione di implementare il metodo Stampa()

Esegui poi le seguenti operazioni:

1. Stampare tutti i treni, in base alla durata del tragitto
2. Stampare tutte le stazioni raggiungibili da Milano entro un'ora
3. Stampare il numero di stazioni presenti in listaTreni, ed infine stampare tutte le stazioni ordinate in base al nome

Soluzione

7. Files

7.1.1 Scrivi un programma che scriva su un file un elenco di nomi, chiuda il file, lo riapra in lettura lo legga tutto e stampi ciò che ha letto.

7.1.2 Scrivi e usa una funzione che conta il numero di caratteri contenuti in un file

7.1.3 Scrivi una funzione che conta e restituisce il numero di righe di un file.
Scrivi poi un'altra funzione che calcola per ogni riga il numero di caratteri contenuti nella riga. I valori vanno inseriti in un array o un vettore da restituire al main.

1. Variante: invece dei caratteri conta il numero di parole per riga

7.1.4 Scrivi e usa una funzione che legga un file e scriva in un altro file tutte le parole del primo file in ordine alfabetico

7.1.5 Scrivi un programma che legga un file di testo e che scriva in un secondo file, lo stesso testo modificato in modo che tutte le lettere siano minuscole. Il file di testo iniziale generalo tu come vuoi.

7.1.6 Scrivi un programma che legga un file di testo e che aggiunga alla fine di tale file, dopo un paio di righe vuote, lo stesso testo modificato in modo che tutte le lettere siano minuscole. Il file di testo iniziale generalo tu come vuoi. Attento a fare in modo che se il programma viene eseguito più volte, le righe da considerare sono solo quelle iniziali.

7.1.7 Scrivi una funzione parametrica in grado di modificare un file di testo di nome "miofile.txt" letto da disco in modo tale che, se l'ultimo elemento della linea è una virgola, la linea successiva venga eliminata

7.1.8 Scrivi una funzione parametrica in grado di modificare un file di testo di nome "miofile.txt" letto da disco in modo tale che, se l'ultimo elemento della linea è una virgola, la linea successiva venga eliminata

7.1.9 Scrivi un programma che scriva su un file un elenco di nomi (o numeri se preferisci), chieda poi all'utente un nome da cercare nel file e stampi sullo schermo l'esito della ricerca (trovato o no). Le operazioni di scrittura su file devono essere eseguite da una funzione che riceve come parametro la stringa contenente il nome del file. Le operazioni di ricerca devono essere svolte da una funzione che riceve come parametro la stringa contenente il nome del file e la stringa da cercare.

1. Continua l'esercizio aggiungendo una funzione che dato il nome del file e il numero di riga, cerca, se esiste, la riga e la restituisce.

7.1.10 Aggiungi anche la possibilità di chiedere all'utente se vuole aggiungere dei nomi a quelli già presenti (aprendo poi il file in modalità append e aggiungendo i nomi dati dall'utente)

7.2 Che richiedono anche le strutture dati

7.2.1 Scrivi una variante dell'esercizio 6.1.4 in cui i dati sugli esploratori sono contenuti nel seguente file di testo

```
Marco Polo 1254 1324
Cristoforo Colombo 1451 1506
Amerigo Vespucci 1454 1512
Francisco Pizarro 1475 1541
Ferdinando Magellano 1480 1521
Hernan Cortez 1485 1547
Walter Raleigh 1552 1618
Henry Hudson 1570 1611
James Cook 1728 1779
Charles Darwin 1809 1882
Kit Carson 1809 1866
David Livingstone 1813 1873
Charles Foucauld 1858 1916
Ronald Amundsen 1872 1928
Ernest Shakleton 1874 1922
```

7.2.2 Scrivi un programma che legga i dati contenuti in un file di testo (testo riportato di seguito) e li inserisca in opportune strutture dati (richiesta la conoscenza delle struct o delle classi). Di questi dati il programma deve fare le seguenti cose:

1. stamparli sullo schermo
2. ordinarli in ordine decrescente di voto
3. stamparli nuovamente in ordine
4. scriverli in ordine in un secondo file che puoi nominare come vuoi

Dati da mettere nel file di testo:

```
Amici  7.00
Biella 9.00
Brescia    6.00
Carolla    8.00
Cunegatti  7.00
DeBella    4.00
DeVecchi   9.00
Fumagalli  7.00
Galimberti 9.00
Germanò    9.00
Gubellini  9.00
Lepore 5.00
Maconi 6.00
Mariani    8.00
Mattavelli 9.00
Oggiano    4.00
Passoni    5.00
Pastori    4.00
Pirovano   7.00
Rudi   10.00
Russell    6.00
Sogos  4.00
Tezza  6.00
Varisco    8.00
```

7.2.3 Scrivi un programma che legga i dati contenuti in un file di testo (testo riportato di seguito) e li inserisca in opportune strutture dati che rappresentino il registro dei voti per una materia. Il programma deve poi calcolare la media dei voti per ogni singolo studente e aggiungerla alla struttura dati (che quindi conterrà anche una variabile media). Il

programma deve mostrare tutti i dati raccolti e calcolati sullo schermo e salvarli in un secondo file di testo.

```
Amici 7.00 8.00 7.00
Biella 8.00 9.50 7.00
Brescia 5.00 7.50 9.00
Carolla 7.00 8.50 8.50
Cunegatti 7.00 7.00 5.50
DeBella 4.00 4.00 4.50
DeVecchi 8.00 10.00 6.00
Fumagalli 8.50 6.00 4.50
Galimberti 9.00 9.00 9.00
Germanò 8.50 9.00 7.50
Gubellini 8.00 10.00 7.00
Lepore 5.50 4.00 3.00
Maconi 7.50 5.00 7.50
Mariani 8.00 8.00 7.00
Mattavelli 9.00 9.50 8.50
Oggiano 5.00 3.00 3.00
Passoni 5.00 6.00 6.00
Pastori 3.50 5.00 7.00
Pirovano 6.50 7.50 7.50
Rudi 9.50 10.00 10.00
Russell 7.50 4.50 5.50
Sogos 4.00 3.50 3.50
Tezza 7.00 5.50 5.50
Varisco 8.00 9.00 8.50
```

7.2.4 Scrivi un programma che legga i dati contenuti in un file di testo (testo riportato di seguito) e li inserisca in opportune strutture dati che rappresentino il registro dei voti per una materia. Il programma deve poi calcolare la media dei voti per ogni singolo studente e aggiungerla alla struttura dati (che quindi conterrà anche una variabile media). Il programma deve mostrare tutti i dati raccolti e calcolati sullo schermo e salvarli in un secondo file di testo. Questa versione del programma è più difficile della precedente perché si può notare che i cognomi possono essere formati da più parole e sono separati dai voti da un ":", inoltre il numero di voti varia per ogni studente.

```
Amici: 7.00 8.00 7.00
Biella: 8.00 9.50 7.00
Brescia: 5.00 7.50 9.00
Carolla: 7.00 8.50 8.50
Cunegatti: 7.00 7.00
De Bella: 4.00 4.00 4.50 5.00
De Vecchi: 8.00 10.00 6.00
Fumagalli: 8.50 6.00 4.50
Galimberti: 9.00 9.00 9.00
Germanò: 8.50 9.00 7.50
Gubellini: 8.00 10.00 7.00
Lepore: 5.50 4.00 3.00 5.00
Maconi: 7.50 5.00
Mariani: 8.00 8.00 7.00
Mattavelli: 9.00 9.50 8.50
Oggiano: 5.00 3.00 3.00 2.00
Passoni: 5.00 6.00 6.00 7.50
Pastori: 3.50 5.00 7.00 2.00
Pirovano: 6.50 7.50 7.50
Rudi: 9.50 10.00 10.00
Russell: 7.50 4.50
Sogos: 4.00 3.50 3.50 2.00
Tezza: 7.00 5.50 5.50
Varisco: 8.00 9.00 8.50
Pirovano: 6.50 7.50 7.50
Rudi: 9.50 10.00 10.00
Russell: 7.50 4.50 5.50
Sogos: 4.00 3.50 3.50
Tezza: 7.00 5.50 5.50
Varisco: 8.00 9.00 8.50
```

Soluzione

7.2.5 Leggi le informazioni riguardanti una serie di libri da un file (il testo è riportato di seguito) e:

1. carica i dati in un array di libri (definisci un'apposita struttura libro)
2. ordina i libri per data di scrittura e riscrivili in un altro file nello stesso formato del file originale
3. scrivi e usa una funzione che riceve i dati già caricati e che restituisce il libro più recente
4. scrivi e usa una funzione che restituisca il secolo in cui sono stati scritti più libri e il numero di libri scritti in quel secolo

```
Divina Commedia;Dante Alighieri;1321
Promessi Sposi;Alessandro Manzoni;1840
Il Decamerone;Alessandro Boccaccio;1350
L'Orlando Furioso;Ludovico Ariosto;1516
Il fu Mattia Pascal;Luigi Pirandello;1904
Ultime lettere di Jacopo Ortis;Ugo Foscolo;1802
Se questo è un uomo;Primo Levi;1947
Il barone rampante;Italo Calvino;1957
I Malavoglia;Giovanni Verga;1881
Il Principe; Niccolò Macchiavelli;1532
La Gerusalemme Liberata;Torquato Tasso;1581
Cuore;Edmondo De Amicis;1886
Il deserto dei Tartari;Dino Buzzati;1940
La coscienza di Zeno;Italo Svevo;1923
Pinocchio;Carlo Collodi;1883
Il piacere;Gabriele D'Annunzio;1889
Myricae;Giovanni Pascoli;1891
Canti;Giacomo Leopardi;1831
Uno, nessuno e centomila;Luigi Pirandello;1926
Il nome della rosa;Umberto Eco;1980
La casa in collina;Cesare Pavese;1948
Il gattopardo;Giuseppe Tomasi di Lampedusa;1958
I Vicerè;Federico De Roberto;1894
Mastro don Gesualdo;Giovanni Verga;1889
```

Soluzione

7.2.6 Scrivi un programma che sia in grado di leggere da file i dati riportati di seguito e salvarli in opportune strutture dati (punto e triangolo). Il programma deve poi stampare sia su schermo che in un altro file i dati riguardanti ogni triangolo e se esso è scaleno, isoscele o equilatero. Per decidere il tipo di triangolo devono essere usate tre diverse funzioni che controllano ognuna se il triangolo passato come parametro è di uno dei tre tipi richiesti.

```
Input
(1, 0) (3, 3) (-0.6, 3.23)
(1, 0) (1, 4) (-2.46, 2)
(2, 0) (5, 3) (2, 6)
(3, 0) (4.4, 3.3) (2.35, 4.8)
(2, 2) (4, 2) (4, 6)

Output
(1.000000, 0.000000) (3.000000, 3.000000) (-0.600000, 3.230000) - equilatero
(1.000000, 0.000000) (1.000000, 4.000000) (-2.460000, 2.000000) - equilatero
(2.000000, 0.000000) (5.000000, 3.000000) (2.000000, 6.000000) - isoscele
(3.000000, 0.000000) (4.400000, 3.300000) (2.350000, 4.800000) - scaleno
(2.000000, 2.000000) (4.000000, 2.000000) (4.000000, 6.000000) - scaleno
```

8. Strutture dati avanzate

8.1 Dizionari

8.1.1 Scrivi una funzione a cui passare una stringa come parametro, e che restituisca un dizionario rappresentante la "frequenza di comparsa" di ciscun carattere componente la stringa.

Semplicemente, data una stringa "ababcc", otterremo in risultato {"a": 2, "b": 2, "c": 2}!

8.1.2 Scrivi un programma che utilizzi una funzione che, ricevuto un array di numeri interi, restituisce il numero che compare con più frequenza (in caso di parità restituisce il primo numero trovato con la frequenza maggiore). Puoi risolvilo in due modi:

1. Con sole liste (inutilmente complicato e fattibile in vari modi complicati)
2. Con un dizionario

Soluzione

8.1.3 Scrivi e usa una funzione che presa in input una lista contenente interi e stringhe, modifica la lista distruttivamente e restituisce un dizionario. Al termine della funzione dalla lista devono risultare cancellate tutte le stringhe e il dizionario restituito deve contenere come chiavi le stringhe cancellate ciascuna con attributo il numero di volte in cui occorrevano nella lista.

Ad esempio per lista=[1,'a',2,'b','a',8,'d',8] la funzione al termine restituisce il dizionario {'a':2,'b':1,'d':1} e la lista diventa [1,2,8,8]

Soluzione

8.1.4 Data la seguente lista di stringhe che rappresenta un girone di Champions:

```
champions= [
    "Sheriff Tiraspol - Shaktar Donetsk  2 0",
    "Inter - Real Madrid 0 1",
    "Shaktar Donetsk  - Inter 0 0",
    "Real Madrid - Sheriff Tiraspol   1 2",
    "Shaktar Donetsk  - Real Madrid  0 5",
    "Inter - Sheriff Tiraspol  3 1",
    "Real Madrid - Shaktar Donetsk  2 1",
    "Sheriff Tiraspol - Inter  1 3",
    "Inter - Shaktar Donetsk  2 0",
    "Sheriff Tiraspol - Real Madrid  0 3",
    "Shaktar Donetsk  -  Sheriff Tiraspol  1 1",
    "Real Madrid - Inter   2 0"
]
```

Sapendo che i nomi delle squadre sono separati da un trattino.
(e che la vittoria vale tre punti, il pareggio 1 punto e la sconfitta 0 punti)

Usando un Dizionario, stampare per ogni squadra, quanti punti ha fatto

Soluzione

8.1.5 Dato il seguente dizionario:

```
d = {
    "senegal" : ("africa", "francese"), "quebec" : ("nordAmerica", "francese"),
    "stati uniti" : ("nordAmerica", "inglese"), "canada" : ("nordAmerica", "inglese"),
    "brasile" : ("sudAmerica", "portoghese"), "messico" : ("nordAmerica", "spagnolo"),
    "australia" : ("oceania", "inglese"), "haiti" : ("centroAmerica", "francese"),
    "angola" : ("africa", "portoghese"), "ciad" : ("africa", "francese"),
    "filippine" : ("asia", "spagnolo"), "argentina" : ("sudAmerica", "spagnolo"),
    "neoZelanda" : ("oceania", "inglese"), "cuba" : ("centroAmerica", "spagnolo"),
    "mozambico" : ("africa", "portoghese"), "mali" : ("africa", "francese"),
    "nigeria" : ("africa", "inglese"), "martinica" : ("centroAmerica", "francese"),
    "uruguay" : ("sudAmerica", "spagnolo"), "cile" : ("sudAmerica", "spagnolo"),
    "caienna" : ("sudAmerica", "francese"), "perù" : ("sudAmerica", "spagnolo")
}
```

1. stampare tutti i paesi di lingua francese
2. con un ciclo while chiedere all'utente di inserire una nuova nazione (non presente nel dizionario) il suo continente e la sua lingua, uscire dal ciclo quando il paese è corretto, inserire queste informazioni nel dizionario
3. per ogni lingua, stampare il numero di paesi
4. per ogni lingua stampare la lista ordinata dei paesi che parlano appunto quella lingua

8.1.6 Data la seguente lista di numeri

lista = [882, 1024, 917, 777, 89, 7, 274, 112, 898, 666, 76, 75737, 6767, 39]

stampare la cifra più presente

(Risultato: la cifra 7 è presente 12 volte)

8.1.7 (dizionario di dizionario --> il secondo dizionario è in pratica usato come fosse una struct)

```
d = {
    "Atalanta": {"city": "Bergamo", "country": "Italia"},
    "Hellas": {"city": "Verona", "country": "Italia"},
    "Sampdoria": {"city": "Genova", "country": "Italia"},
```

```
    "Genoa": {"city": "Genova", "country": "Italia"},
    "Liverpool": {"city": "Liverpool", "country": "RegnoUnito},
    "Everton": {"city": "Liverpool", "country": "RegnoUnito"},
    "Espanol": {"city": "Barcellona", "country": "Spagna"},
    "Barcellona": {"city": "Barcellona", "country": "Spagna"},
    "Siviglia": {"city": "Siviglia", "country": "Spagna"},
    "Betis": {"city": "Siviglia", "country": "Spagna"},
    "Juventus": {"city": "Torino", "country": "Italia"},
    "YoungBoys": {"city": "Berna", "country": "Svizzera"},
    "Chievo": {"city": "Verona", "country": "Italia"}
}
```

1. Trovare la città dell'Hellas, e stampare il nome dell'altra squadra della stessa città
2. stampare in ordine alfabetico tutte le squadre italiane

8.1.8 Riprendendo il dizionario dell'esercizio precedente:

Stampare le sole squadre italiane, ordinate in base alla città, e a parità di città in ordine alfabetico

```
d = {
    "Atalanta" : {"city": "Bergamo", "country": "Italia"  },
    "Hellas" :   {"city": "Verona", "country": "Italia"  },
    "Sampdoria" : {"city": "Genova", "country": "Italia"  },
    "Genoa" :   {"city": "Genova", "country": "Italia"  },
    "Liverpool" : {"city": "Liverpool", "country": "RegnoUnito" },
    "Everton" : {"city": "Liverpool", "country": "RegnoUnito"  },
    "Espanol" : {"city": "Barcellona", "country": "Spagna"  },
    "Barcellona" : {"city": "Barcellona", "country": "Spagna"  },
    "Siviglia" : {"city": "Siviglia", "country": "Spagna"  },
    "Betis" :   {"city": "Siviglia", "country": "Spagna"  },
    "Juventus" :   {"city": "Torino", "country": "Italia"  },
    "YoungBoys" :   {"city": "Berna", "country": "Svizzera"  },
    "Chievo" :   {"city": "Verona", "country": "Italia"  }
}
```

Soluzione

8.1.9 Dato il seguente dizionario di dizionario:

```
d = {
    "A1" : {"percorso":"Milano-Napoli", "gestore":"Autostrade per l'Italia", "km":759},
    "A2" : {"percorso":"Salerno-ReggioCalabria", "gestore":"Anas", "km":442},
    "A3" : {"percorso":"Napoli-Salerno", "gestore":"Autostrade per l'Italia", "km":52},
```

```
    "A4" : {"percorso":"Torino-Trieste", "gestore":"Autostrade per l'Italia", "km":524},
    "A6" : {"percorso":"Torino-Savona", "gestore":"Gruppo Gavio", "km":124},
    "A7" : {"percorso":"Milano-Genova", "gestore":"Enti Pubblici", "km":133},
    "A8" : {"percorso":"Milano-Varese", "gestore":"Autostrade per l'Italia", "km":43},
    "A10" : {"percorso":"Savona-Ventimiglia", "gestore":"Gruppo Gavio", "km":113},
    "A11" : {"percorso":"Firenze-Pisa", "gestore":"Autostrade per l'Italia", "km":81},
    "A12" : {"percorso":"Genova-Cecina", "gestore":"Gruppo Gavio", "km":210},
    "A13" : {"percorso":"Bologna-Padova", "gestore":"Autostrade per l'Italia", "km":116},
    "A14" : {"percorso":"Bologna-Taranto", "gestore":"Autostrade per l'Italia", "km":743},
    "A15" : {"percorso":"Parma-LaSpezia", "gestore":"Gruppo Gavio", "km":108},
    "A16" : {"percorso":"Napoli-Canosa", "gestore":"Autostrade per l'Italia", "km":172},
    "A19" : {"percorso":"Palermo-Catania", "gestore":"Anas", "km":191},
    "A21" : {"percorso":"Torino-Piacenza-Brescia", "gestore":"Gruppo Gavio", "km":238},
    "A22" : {"percorso":"Brennero-Modena", "gestore":"Enti Pubblici", "km":315},
    "A23" : {"percorso":"Palmanova-Tarvisio", "gestore":"Enti Pubblici", "km":119},
    "A24" : {"percorso":"Roma-Teramo", "gestore":"Gruppo Toto", "km":159},
    "A25" : {"percorso":"Torano-Pescara", "gestore":"Gruppo Toto", "km":115},
    "A26" : {"percorso":"Genova-Gravellona", "gestore":"Autostrade per l'Italia", "km":197},
}
```

1. Stampare il gestore dell'autostrada A8
2. stampare il gestore dell'autostrada "Savona-Ventimiglia"
3. stampare l'autostrada più lunga, con il suo percorso e il suo gestore
4. Stampare quanti differenti gestori sono presenti nel dizionario (per la soluzione utilizzare una semplice lista)
5. Per ognuno dei gestori presenti stampare il numero di autostrade gestite, e il numero di km gestiti

Soluzione

8.1.10 Il seguente dizionario descrive quante ore hanno lavorato in una settimana, una serie di dipendenti: ad esempio Me White ha lavorato 8 ore il lunedì, 9 ore il martedì, 8 ore il mercoledì, 9 ore il giovedì e 4 ore il venerdì

```
d2 = {
    "Mr White": [8, 9, 8, 9, 4],
    "Mr Brown": [0, 8, 7, 10, 8],
    "Mr Blonde": [8, 8, 8, 9, 9],
    "Mr Black": [6, 9, 9, 8, 8],
    "Mr Red": [ 8, 7, 8, 8, 8],
    "Mr Green": [ 4, 8, 8, 8, 4]
}
```

1. Stampare la classifica dei lavoratori, in base al numero di ore lavorate
2. Stampare il giorno con più ore lavorate, e quante ore sono state lavorate in quel giorno

Soluzione

8.1.11 progettare la funzione f(ftesto,k) che, presi in input il nome (o l'indirizzo) di un file di testo ed un intero k, restituisce una stringa di caratteri lunga k. Il file di testo contiene stringhe di diversa lunghezza (una per riga ed ogni riga termina con '\n'), un possibile testo da inserire nel file di input è riportato di seguito.

I k caratteri della stringa restituita dalla funzione si ottiengono considerando le stringhe lunghe k presenti nel file di testo. L'i-mo carattere della stringa sara' il carattere che compare con maggior frequenza come i-mo carattere delle stringhe lunghe k nel file di testo (in caso di parita' di occorrenze viene scelto il carattere che precede gli altri lessicograficamente).

Nel caso il file di testo non contenga parole lunghe k allora viene restituita

la stringa vuota.

Ad Esempio, per il file di testo:

```
porta
rigore
ora
giacca
follia
gara
finale
salute
fiore
lampada
cane
erba
palo
tre
due
fionda
limite
polo
soglia
amo
```

e k=3 la funzione restituisce la stringa 'are' a seguito della presenza delle seguenti 4 stringhe lunghe 3:

```
tre
due
amo
ora
```

Soluzione

9. Altro

9.1 Sistema

Questo tipo di programmi non rientra nel normale percorso di studio ma siccome è utile e interessante lo aggiungo per gli interessati. Parte degli esercizi con le relative soluzioni sono presi da https://www.programmareinpython.it/esercizi-python/

9.1.1 Scrivi una funzione che fornisca in output il nome del Sistema Operativo utilizzato con eventuali relative informazioni sulla release corrente.

Soluzione

9.1.2 Scrivi una funzione che calcoli la somma (espressa in MB) delle dimensioni dei file presenti nella cartella di lavoro.

Soluzione

9.1.3 Scrivi una funzione "postino" che sia in grado di spedire delle eMail tramite Gmail! (aiuto: puoi usare il modulo smtplib)

Soluzione

9.1.4 Scrivi una funzione "cercatrice" che scansioni un dato percorso di sistema alla ricerca di file di tipo pdf.

La funzione dovrà avere le seguenti caratteristiche:

Il percorso fornito dovrà essere anzitutto validato, in quanto deve portare a una cartella esistente

La funzione dovrà fornire un elenco dei file pdf (con/relativo/percorso) man mano che questi vengono trovati

In fine la funzione dovrà fornire in output il totale dei file .pdf che sono stati trovati durante la scansione.

Soluzione

9.1.5 Scrivi una funzione "file_backup" che sia in grado di effettuare copie di backup di determinati tipi di file, con le seguenti caratteristiche:

Percorso da scansionare, di backup e tipologia di file da copiare dovranno essere passati dall'utente tramite input

Il programma dovrà verificare la presenza o meno di una cartella di backup al percorso fornito, e qualora questa non fosse presente dovrà crearla

La funzione dovrà anche gestire l'eventuale scelta da parte dell'utente, di un percorso da scansionare che non esiste

Soluzione