

DOMANDE ARCHITETTURE DEGLI ELABORATI (video 21/22)

DOMANDA N°1: Descrivere le funzionalità della memoria RAM, le sue caratteristiche e le principali differenze rispetto ad un hard-disk allo stato solido (SSD).

Memoria RAM:

- Volatile
- Veloce (circa 10ns)
- Interna
- Abbastanza capiente (4-8-16 GB)
- Più costosa
- La CPU può accedere direttamente (per leggere/scrivere dati o per eseguire programmi)

Memoria hard-disk (SSD):

- Permanente
- Lenta (circa 10ms)
- Esterna
- Molto capiente (1-2 TB = 1000/2000 GB)
- Più economica

DOMANDA N°2: Qual è la differenza tra un bus seriale ed uno parallelo? Fare un esempio di bus seriale ed uno di bus parallelo.

Entrambi i bus sono strutture di collegamento per il trasferimento dati. Un bus seriale fa passare 1 bit alla volta, è composto solo di 2 fili, di cui 1 per i dati (esempio: bus USB, bus HDMI). Un bus parallelo fa passare N bit per volta, è composto di N+1 fili (esempio: vecchi collegamenti alle stampanti o bus interni dell'elaboratore tipo bus di dati o indirizzi).

DOMANDA N°3: Si descriva cosa si intende per 'gerarchia di memoria' in un sistema di elaborazione, e quali siano le caratteristiche fondamentali che differenziano i vari tipi di memorie.

DOMANDA N°4: Spiegare brevemente cosa si intende per gerarchia di memoria in un calcolatore e quali sono i vantaggi che derivano da tale struttura.

Ci sono memorie diverse:

- Le memorie più vicine alla CPU sono più veloci ma meno capienti (viva via che mi allontanano dalla CPU le memorie saranno più capienti, ma più lente e possono essere permanenti (non volatili)). Non si possono avere tutte veloci per ragioni di costo.
CPU: registri -> cache -> RAM -> SSD -> Hard Disk (ordine di vicinanza, dalla più vicina alla più lontana).
Tutto il meccanismo serve per far arrivare alla CPU i dati alla velocità massima, ma permettendo di accedere alla capienza massima.

Caratteristiche: capienza, velocità, costo.

Registri, cache, RAM: memorie volatili

SSD, Hard Disk: memorie permanenti

DOMANDA N°5: Descrivere l'unità di controllo, indicando da quali blocchi è composta e quali sono le loro funzioni.

La CPU è fatta da unità di controllo e unità operativa.

L'unità di controllo (UC) decide quali operazioni eseguire e, quindi, contiene al suo interno:

- PC (Program Counter): memorizza l'indirizzo della prossima istruzione che la CPU dovrà eseguire
- IR (Instruction Register): codifica binaria (letta dalla memoria all'indirizzo indicato dal PC) dell'istruzione da eseguire
- Logica di decodifica, dal contenuto dell'IR determina i comandi da dare alla UO ed alla memoria.

PC e IR permettono alla CPU di reperire, una dopo l'altra, tutte le istruzioni che compongono il programma da eseguire.

L'unità operativa (UO) contiene:

- ALU unità aritmetico logica (FPU floating point unit)
- Registri
- Registro dei Flag

DOMANDA N°6: Descrivere le funzioni e le caratteristiche dei bus presenti in un tipico sistema di elaborazione.

I bus sono una infrastruttura di comunicazione interna all'elaboratore tra i diversi componenti di un sistema di elaborazione, per trasferire i dati della CPU da/verso la memoria o le periferiche.

- Bus indirizzi (ABUS): bus parallelo di N bit che specifica l'indirizzo di memoria da cui leggere/scrivere
- Bus dati (DBUS): bus parallelo di M bit sui cui trasferiamo da un componente ad un altro i dati letti/scritti
- Bus di controllo (CBUS): gestiscono i comandi e le temporizzazioni delle attività.

Capacità di memoria indirizzabile dalla CPU: M bit x 2^N indirizzi

- potrà leggere tanti dati quante sono le celle di memoria indirizzabili (2^N)
- dimensione della singola cella di memoria = M (ampiezza DBUS)

DOMANDE ARCHITETTURE DEGLI ELABORATI (video 22/23)

DOMANDA N°1: Che cosa sono e qual è la funzione dei registri di una CPU? A cosa serve il registro dei flag?

I registri sono delle celle di memoria all'interno dell'unità operativa della CPU. Servono a memorizzare i dati sui cui la ALU deve lavorare (gli operandi) ed i risultati calcolati dalla ALU stessa (idem per la FPU).

Il registro dei flag memorizza una serie di bit 'singoli', ciascuno dei quali indica un aspetto dell'ultima operazione eseguita dall'ALU/FPU (zero, negativo, overflow...)

DOMANDA N°2: Spiegare brevemente cos'è il clock di sistema e qual è la sua unità di misura. / Spiegare qual è la funzione del clock. Cosa si intende per periodo e per frequenza del clock?

Il clock di sistema è un segnale periodico presente all'interno dell'elaboratore, che serve a temporizzare le operazioni interne alla CPU. Il suo scopo è fornire la temporizzazione per le diverse attività interne alla CPU. I dati vengono letti all'inizio di un ciclo di clock ed i risultati sono disponibili entro la fine del ciclo. Ogni operazione dell'elaboratore avviene in corrispondenza di un istante di clock. È caratterizzato da un periodo T (in us o ns) o da una frequenza f (in MHz o GHz) con $T = 1/f$ che governa le operazioni interne.

LA frequenza di clock può essere aumentata o ridotta in funzione del carico di lavoro del sistema e/o delle necessità di risparmio energetico.

DOMANDA N°3: Qual è il numero minimo di bit di Address Bus necessario per indirizzare una memoria da 4 megabyte con un parallelismo di 16 bit?

UNICA FORMULA:

numero di bit = numero di celle x dimensione di una cella di memoria

massimo numero di celle = $2^{|ABUS|}$

dimensione di una cella = $|DBUS|$ (dimensione di un databus)

la quantità di memoria massima (in bit) non potrà mai superare il numero di celle massime che posso utilizzare ($2^{|ABUS|}$) moltiplicato per la dimensione di una cella ($|DBUS|$)

$MEM\ MAX = 2^{|ABUS|} \times |DBUS|$

$MEM = 4\ MB$

Dimensione cella = 16 bit = 2 B

$4\ MB / 2\ B = 2\ M = 2 \times 2^{20} = 2^{21}$ celle

DOMANDA N°4: Qual è il ruolo della Floating Point Unit in una CPU?

La FPU fa parte dell'unità operativa della CPU e contiene circuiti logici in grado di svolgere le operazioni aritmetiche (+-*) e matematiche (exp, radici, trig...) su numeri rappresentati in virgola mobile (floating)

point), single o double precision. Operazioni tra i valori contenuti nei registri, su ordine dell'unità di controllo.

Si distingue dalla ALU che ha lo stesso scopo ma lavora sui numeri interi.

Il tempo necessario per svolgere un'operazione in virgola mobile è decisamente superiore rispetto ad un'operazione tra numeri interi.

DOMANDA N°5: Data una memoria RAM di dimensione 64kbyte e, con Address Bus su 12 bit, se ne calcoli la dimensione di ogni cella.

$$64 \text{ KB} / 2^{12} = 2^6 \times 2^{10} \text{ (K)} \times 2^3 \text{ (B)} / 2^{12} = 2^7 = 128 \text{ bit}$$

DOMANDA N°6: Descrivere come si compone il ciclo macchina di un elaboratore.

Il ciclo macchina di un elaboratore si compone di tre fasi, con cui la CPU esegue le istruzioni dei programmi, che si ripetono all'infinito: Fetch, Decode, Execute.

FETCH: la CPU legge dalla memoria RAM il codice della prossima istruzione da eseguire (l'indirizzo della cella di memoria è nel registro PC – program counter), il codice dell'istruzione viene salvato nel registro IR = MEM(PC).

DECODE: Sulla base del contenuto del IR (istruzione da eseguire) appena ricevuto, lo analizza e 'capisce' quali siano le operazioni da svolgere (si definiscono i Comandi che l'UC invierà alla UO).

EXECUTE: la UO, la MEM e le altre componenti coinvolte eseguono i comandi, inviati dalla UC, che sono stati decisi nella fase di Decode. La UC attende il loro completamento.

DOMANDA N°7: Spiegare brevemente la funzione dei registri presenti nell'Unità Operativa di una CPU.

(vedi domanda n°1).

DOMANDA N°8: Spiegare perché i computer attuali sono quasi tutti a 64 bit. Cosa significa 'a 64 bit'?

'64 bit' rappresenta la dimensione dei dati su cui lavora la CPU: la dimensione dei registri, la dimensione della ALU, parallelismo della memoria, dimensione del DBUS.

64 bit è la dimensione 'naturale' dei dati trattati (la dimensione di un 'int').

64 bit è un compromesso tra le capacità di calcolo (più bit = più potenza) e il costo realizzativo (più bit = più area di silicio delle CPU e memorie = più costi).

DOMANDA N°9: Illustrare quali sono le principali unità del microprocessore. Cosa è la frequenza di clock?

Un microprocessore è fatto dall'unità centrale (CPU) che è divisa in UC (PC e IR) e UO (registri, ALU, FPU) e può contenere anche la memoria cache di primo livello L1 che è esterna alla CPU, a volte ci sono anche altri blocchi logici/altri periferici (USB ecc...).

DOMANDA N°11: Data una memoria RAM di dimensione 4Mbyte con Address bus su 16 bit, se ne calcoli il parallelismo. Si riportino tutti i passaggi per la soluzione.

MEM = 4MB

ABUS = 16 bit = 2^{16} celle

Dimensione cella = MEM/n celle = $4 \text{ MB} / 2^{16} = (2^2 \times 2^{20} \times 2^3) / 2^{16} = 2^9 = 512 \text{ bit}$

DOMANDA N°12: Cos'è la ALU?

La ALU fa parte dell'unità operativa della CPU e contiene circuiti logici in grado di svolgere le operazioni aritmetiche (+*;) e logiche (and/or/not...) su numeri rappresentanti in binario puro o complemento a due.

DOMANDA N° 13: Si supponga di avere un address bus (ABUS) da 20 bit. Indicare la massima quantità di memoria fisica che è possibile installare nell'elaboratore nel caso in cui la singola cella di memoria abbia ampiezza 4 byte.

ABUS = 20 bit

Cella = 4 byte

MEM = $2^{|ABUS|} \times \text{cella} = 2^{20} \times 2^2 = 2^{22} \text{ B} = 4 \text{ MB} = 32 \text{ Mb}$

DOMANDA N°14: Descrivere il concetto di gerarchia della memoria nelle moderne architetture di elaborazione.

Nell'elaboratore abbiamo più memorie con caratteristiche diverse:

- Più veloci (e costose) ma meno capienti, più vicine alla CPU
- Più lente (e meno costose) e più capienti, più lontane della CPU

Registri, Cache L1, Cache L2, RAM, memoria di massa (SSD, Hard Disk).

DOMANDA N°15: Descrivere le funzioni e le caratteristiche dei bus presenti in un tipico sistema di elaborazione.

In un sistema di elaborazione abbiamo 3 bus:

- Address Bus: indica gli indirizzi di memoria (indirizzi dei periferici) da cui leggere o scrivere un dato. È impostato dall'unità di controllo.
- Data Bus: contiene il dato letto dalla memoria (o da un periferico di input) o il dato da scrivere in memoria (su un periferico di output). Può essere impostato dalla UO (scrittura) oppure dalla RAM (o periferico) (lettura).
- Control Bus: contiene una serie di segnali di 'comando' (la UC ordina l'operazione da fare, la tempistica, lo stato dell'operazione eseguita).

DOMANDA N°16: Descrivere l'unità di controllo, indicando da quali blocchi è composta e quali sono le loro funzioni.

L'unità di controllo è una parte della CPU insieme alla UO. L'unità di controllo contiene:

- Il Program Counter
- L'Instruction Register
- Dei comandi che vanno all'unità operativa che sono generati da una logica di decodifica che prende l'IR e lo trasforma in comandi. Questa logica riceve informazioni di stato dall'UO che sono i Flag.
- Interfaccia verso la memoria (verso i bus) perché nella fase di Fetch devo mandare il Program Counter all'address bus e ricevere l'istruzione dalla memoria. L'unità di controllo deve avere la capacità di comandare un'operazione di lettura della memoria Ram che sarà sul bus.

DOMANDA N°17: Si descriva cosa si intende per 'gerarchia di memoria' in un sistema di elaborazione, e quali siano le caratteristiche fondamentali che differenziano i vari tipi di memorie.

Nell'elaboratore abbiamo più memorie con caratteristiche diverse:

- Più veloci (e costose) ma meno capienti, più vicine alla CPU
- Più lente (e meno costose) e più capienti, più lontane della CPU
- Volatilità (memorie che perdono i loro dati quando manca l'alimentazione elettrica; tutte le memorie interne sono volatili, le memorie di massa (dischi) non sono volatili)

Registri, Cache L1, Cache L2, RAM, memoria di massa (SSD, Hard Disk).

DOMANDA N°18: Per quale motivo la velocità di clock nelle CPU è andata sempre crescendo negli ultimi quarant'anni? Perché negli ultimi anni questa crescita si è fortemente ridotta?

La velocità di clock è sempre andata aumentando per costruire sistemi di elaborazione a maggiori prestazioni per necessità di sempre maggiore potenza di calcolo, grazie al miglioramento tecnologico (transistor, porte logiche, memorie) si possono costruire circuiti più veloci.

Negli ultimi anni per motivi fisici e tecnologici la frequenza di clock non è aumentata oltre ai 3-4 GHz. L'aumento delle prestazioni si è ottenuto con mezzi diversi, come CPU multiple (es. multi-core).

DOMANDA N°19: Qual è la differenza tra un bus seriale ed uno parallelo) Fare un esempio di bus seriale ed un esempio di bus parallelo.

Un bus seriale (USB) è un bus fatto da un filo solo. Un bus parallelo (più veloce, più costoso, più ingombrante) è composto da più fili che viaggiano in parallelo (solitamente bus interni ad un elaboratore).

DOMANDE ARCHITETTURE DEGLI ELABORATI (video 23/24)

DOMANDA N°1: Descrivere il ruolo dell'unità operativa e dei suoi componenti.

Componenti:

- Registri (tra cui registro dei flag)
- Unità aritmetico logica
- Unità floating point

Ruolo:

- Eseguire le operazioni di lettura. Scrittura, calcolo aritmetico ordinate dall'UC e salvarne i risultati. I dati e i risultati su cui lavora l'UO sono memorizzati nei registri.

DOMANDA N°2: Calcolare la massima dimensione di memoria allocabile in un sistema con un Address Bus di 24 bit e parallelismo di 16 byte. Esprimere il risultato nel massimo multiplo di byte possibile, mantenendo il valore comunque maggiore o uguale ad 1 (esempio: 128 KB, non 0.1 MB).

ABUS di 24 bit -> ogni indirizzo di memoria è un intero su 24 bit, quindi potrò avere massimo 2^{24} celle di memoria.

Ciascuna cella contiene 16 byte.

MEMMAX = max n.celle * dimensione di ciascuna cella

MEMMAX = $2^{24} * 16$ byte =

2^{28} byte =

$2^{20} * 2^8$ byte = MEGA BYTE * 2^8

256 MB poiché $2^{20} = 1$ MB

DOMANDA N°3: Quali sono le funzioni della ALU?

ALU è un circuito in grado di calcolare tutte le operazioni possibili sui numeri interi (CA2): somme, sottrazioni, moltiplicazioni, (talvolta divisioni e modulo), confronti, operazioni booleane sui bit (and, or, not).

Le operazioni sono svolte prendendo in ingresso i valori dai registri e depositando il risultato in un altro registro, contemporaneamente aggiornando il registro dei Flag. L'operazione svolta dall'ALU e i registri coinvolti sono decisi dall'unità di controllo.

DOMANDA N°4: Concetto gerarchia della memoria.

In un sistema di elaborazione ci sono varie tipologie di memorie, che variano in funzione della velocità di accesso, del costo, della capienza, della persistenza. ALL'aumentare della velocità aumentano i costi e diminuisce la capienza. Nel sistema di elaborazione si organizza una 'catena' di memorie, partendo da quelle più veloci che sono poste vicino alla Cpu, ed allontanandosi memorie via via più capienti e più lente.

Un esempio di gerarchia:

- Registri della UO > Cache di 1 livello > Cache di secondo livello > Memoria principale > Memorie di massa (SSD, HD, Hard Disk).

DOMANDA N°5: Che cos'è la RAM (memoria principale/primaria)? A cosa serve?

La RAM è la memoria centrale dell'elaboratore. (RAM = Random Access Memory). Essa contiene:

- Il programma da eseguire
- I dati di input al programma
- I risultati calcolati dal programma.

La CPU ha accesso diretto ai contenuti della RAM.

DOMANDA N°6: Quali sono le funzioni dell'unità di controllo?

UC è all'interno della CPU. Funzioni:

- Controllare l'esecuzione dei programmi gestendo il ciclo di esecuzione delle istruzioni.
- Invia i comandi a tutti i componenti del sistema
- Riceve le informazioni di stato da tutti i componenti del sistema
- Gestisce anche la temporizzazione delle varie operazioni, e dell'accesso al bus.

DOMANDA N°7: La memoria RAM di un sistema di elaborazione ha un parallelismo di 8 bit. Avendo a disposizione 1024 celle di memoria, quanti numeri rappresentati in virgola mobile (double precision, 64 bit) è possibile memorizzare in tali celle?

RAM = 8 bit

1024 celle di memoria

1 numero = 64 bit = 8 byte

RAM da 1024 byte

$1024 \text{ B} / 8 \text{ B} = 128$ numeri

DOMANDA N°8: Indicare quali sono le caratteristiche di un moderno pc portatile, in termini di velocità di clock, memoria RAM, hard disk, display.

CPU clock: 2.x – 3.x GHz

MEM centrale: 16 GB – 32 GB

MEM massa: 512 GB SSD – 1 TB o 2 TB (SSD o HD)

Display risoluzione: 4k (3000x2000) minimo (1920x1080) pollici

DOMANDE PROGRAMMAZIONE PYTHON (video 21/22)

DOMANDA N°1: Spiegare il concetto di alias, in Python ed i potenziali problemi ad esso correlati durante la scrittura del codice.

Si dice alias una variabile che faccia riferimento allo stesso valore di un'altra variabile. Lo stesso valore può essere letto/modificato attraverso l'una o l'altra delle variabili alias. Una funzione può modificare un contenitore definito in un'altra funzione, perché gli argomenti di una funzione possono essere alias di variabili di altre funzioni. Modificando il valore tramite una delle variabili alias, la modifica risulta 'visibile' anche tra l'altro alias (se non voglio ciò, devo fare una copia e non un alias). Ad esempio:

```
a = [1, 2, 3]
```

```
b = a #b è un alias
```

I problemi che possono esserci sono di due tipi:

- Le modifiche fatte ad a influenzano anche b, e viceversa
 - Nelle funzioni bisogna fare attenzione a non perdere l'alias qualora la funzione debba modificare il valore passato
- ```
B = b + 4 # b non è più un alias ma punta ad una nuova lista
b.extend(4) #viene estesa la stessa lista di prima
```

**DOMANDA N°2: Perché i dizionari sono chiamati così?**

Dizionario mette in relazione una 'chiave' con un 'valore', come in un dizionario in cui a ciascuna parola (chiave) associo una descrizione/spiegazione (valore). Al dizionario si fa sempre accesso per mezzo di una chiave.

**DOMANDA N°3: Qual è lo scopo principale dell'indentazione nel linguaggio python? Fare un esempio in cui essa risulti di vitale importanza.**

L'indentazione serve per delimitare i 'blocchi' per tutte le istruzioni composte. ES: nei costrutti if/while/for/with/def/try ecc...

If a < 0:

```
 print(a)
```

```
print(b) #viene stampata sempre
```

if a < 0:

```
 print(a)
```

```
 print(b) #stampata solo se a < 0
```

**DOMANDA N°4: Fornire un esempio in cui l'ordine delle istruzioni elif influenzi il risultato finale.**

(riguarda video).

**DOMANDA N°5: Spiega il concetto di visibilità di una variabile e indicare la differenza fra variabili locali e variabili globali.**

La visibilità di una variabile è la porzione di codice in cui tale variabile può essere utilizzata (dal punto in cui viene definita fino alla fine della funzione, oppure fino alla fine del file se è definita fuori da una funzione).

Le variabili locali sono definite solo all'interno di una funzione, visibili solo dal codice della funzione stessa.

Le variabili globali sono definite all'esterno delle funzioni e sono visibili a tutto il codice del programma (interno ed esterno alle funzioni). Usando l'istruzione `global` è possibile rendere globale una variabile definita dentro una funzione.

È preferibili usare variabili locali perché non si rischiano di modificare dati non desiderati.

**DOMANDA N°6: Fornire un esempio di dati per la cui rappresentazione è più opportuno utilizzare una lista e dati per la cui rappresentazione è più opportuno utilizzare un insieme.**

Un elenco alfabetico di nomi di persona (per una lista).

Numeri usciti all'otto (per un set).

**DOMANDA N°7: Quali tra le seguenti istruzioni permette di creare una variabile `d2` che contenga una copia del dizionario `d1`?**

1. `d2 = d1`
2. `d2 = dict(d1.values())`
3. `d2 = dict(d1.keys())`
4. `d2 = dict(d1)`

- 1) No, perché è un alias
- 2) No, perché impossibile creare un dizionario da una lista e mancherebbero le chiavi
- 3) No, perché impossibile creare un dizionario da una lista e mancherebbero i valori
- 4) Sì

**DOMANDA N°8: Confronto tra dizionario e insieme. Indicare almeno una caratteristica comune.**

Dizionario:

- Contiene delle associazioni chiave-valore
- È mutabile (posso aggiungere/cancellare chiavi e modificare valori)
- Le chiavi sono univoche e non duplicate (caratt. In comune: le chiavi di un dizionario si comportano come un insieme perché univoche e immutabili)
- Le chiavi sono in ordine di inserimento

Insiemi:

- Contiene valori singoli (immutabili)
- È mutabile (posso aggiungere/cancellare elementi)
- Gli elementi sono univoci e non duplicati (caratt. In comune)
- Gli elementi non sono ordinati

**DOMANDA N°9: Quali di queste strutture non si può realizzare in python e perché**

**1) Dizionario di liste**

Sì, dizionario i cui valori (non le chiavi) sono liste

**2) Set di dizionari**

No, perché un dizionario non è immutabile, quindi non si può mettere in un set

**3) Dizionario di set**

Sì, dizionario in cui i valori (non le chiavi) siano dei set

**4) Lista di liste**

Sì

## DOMANDE PROGRAMMAZIONE PYTHON (video 23/24)

### DOMANDA N°1: Come si può determinare se una stringa sia trasformabile in intero?

Una stringa è trasformabile in int se contiene la corretta codifica di un numero intero, e si può convertire con la funzione int(s).

Ho due possibilità:

- Analizzare la stringa e verificare i caratteri contenuti.
- Provare a fare la conversione e controllare se si verifica un'eccezione.

Es: try:

```
n = int(s)
```

except NumberFormatException:

```
print('Non convertibile')
```

```
ok = False
```

### DOMANDA N°2: Nella gestione delle eccezioni, quando è eseguita la clausola except? Aggiungere un (breve) esempio esplicativo in Python.

La except viene eseguita quando una delle istruzioni nel blocco try genera un'eccezione del tipo specificato nella Except

Try:

```
f = open('acb.txt', 'r') #catturata – vado in except
```

```
x = 1/0 # eccezione non catturata, termina il programma
```

except OSError:

```
#file non trovato
```

### DOMANDA N°3: Quali sono le caratteristiche principali degli insiemi in python? Quali sono le differenze fondamentali tra liste e insiemi?

Set() è un contenitore mutabile ( posso aggiungere/togliere elementi), i valori contenuti devono essere immutabili (stringhe, numeri, tuple) e confrontabili. Non ammettono duplicati. Posso compiere le normali operazioni insiemistiche. Non è ordinato (l'iterazione avviene in un ordine arbitrario).

È possibile convertire una lista in un insieme: funziona se gli elementi di x sono immutabili, e cancella tutti i duplicati eventualmente presenti. E' possibile convertire un insieme in una lista: funziona sempre e assegna un ordine arbitrario agli elementi.

Differenze con liste:

- le liste sono ordinate, set non ordinati

- liste possono avere elementi duplicati, set no
- liste contengono qualsiasi elemento, set solo elementi immutabili
- i set sono molto ottimizzati nelle operazioni di ricerca.

Vantaggi/Svantaggi:

- il set non ha duplicati, la lista può avere duplicati
- il set non mantiene l'ordine, la lista mantiene l'ordine di inserimento
- nei set gli elementi non hanno indice, non posso accedere al singolo; nelle liste posso accedere al singolo elemento conoscendo l'indice
- nei set è rapidissimo controllare se un elemento esiste, nelle liste il controllo di esistenza richiede di leggere tutti gli elementi
- Il set può contenere solo dati immutabili, la lista può contenere qualsiasi tipo di dato

**DOMANDA N°4: Descrivere il costrutto try/except e motivare con un brevissimo esempio.**

#vedi n°2

**DOMANDA N°5: In una funzione qual è la differenza tra argomenti e valori restituiti? Quanti argomenti può avere una invocazione di funzione? E quanti valori può restituire?**

Def funzione (argomento1, argomento2, ecc...)

....

Return risultato

Gli argomenti sono riferimenti a valori provenienti dalla chiamante e inviati alla funzione, sotto forma di 'alias' degli oggetti posseduti dal chiamante. Una funzione può avere un numero qualsiasi di argomenti; alcuni argomenti possono essere opzionali, se viene fornito un valore di default.

Il risultato è il valore calcolato dalla funzione e che viene trasferito al chiamante, come alias di un valore locale (il quale viene dimenticato quando la funzione termina). Il risultato è uno solo, ma se servisse trasferire più valori si può costruire e restituire una tupla(o lista) contenente i valori richiesti.

**DOMANDA N°6: È possibile usare una lista come chiave di un dizionario Python? E una tupla? Rispondere sì o no, motivando la risposta.**

La chiave di un dizionario deve essere un tipo di dato immutabile, quindi lista no, ma tupla sì.

**DOMANDA N°7: Come si può fare in Python a convertire una stringa in maiuscolo?**

Dat una stringa s, non posso convertirla in maiuscolo (perché è immutabile), ma è possibile costruire una nuova stringa che contenga la versione minuscola/maiuscola, usando il metodo s = s.upper() \*in minuscolo s = s.lower()

**DOMANDA N°8: Cosa succede se dopo aver costruito l'insieme Python 'colori'? Come segue:**

`'colori = set(['bianco', 'giallo', 'verde', 'rosso', 'viola'])'` viene eseguita la riga di codice `'print(colori[2])'`

L'istruzione genera eccezione perché colori è un insieme, e non è possibile accedere per 'posizione' agli elementi di un insieme. (Prova sulla console che tipo di eccezione genera).

**DOMANDA N°9: Quali sono gli operatori booleani? Scrivere un'istruzione Python contenente un operatore booleano.**

Gli operatori booleani sono quelli che fanno calcoli con gli operatori booleani True e False: and, or, not. Li usiamo nelle condizioni complesse, per manipolare flag:

`if a < 0 and a != 7.`

**DOMANDA N°10: Nel linguaggio Python cosa si intende quando si parla della differenza tra la copia di una lista e un alias alla lista stessa? Fornire un esempio.**

Data una lista `A = [1,2,3]` si intende:

- Copia di A: nuova variabile che fa riferimento ad una nuova lista che contiene gli stessi elementi  
`B = list(A), B = A[:], B = A.copy()`
- Alias di A: nuova variabile che fa riferimento alla stessa lista di A  
`B = A`, passaggio di parametri a una funzione

Modificando una copia di una lista la lista originale non viene modificata. Modificando un alias di una lista, in realtà si modifica anche la lista originale.

Esempio: `B.append(7)`

**DOMANDA N°11: Spiegare la differenza tra liste e tuple. Una funzione può restituire una tupla? E una lista?**

Tuple e liste sono analoghe (contenitori ordinati di valori qualsiasi, posso accedere per posizione e per range). La differenza è che le tuple non sono mutabili, mentre le liste sono mutabili.

Una funzione può restituire qualunque tipo di oggetto in generale, quindi, sicuramente, una lista e anche una tupla.

**DOMANDA N°12: Come si esegue un controllo di flusso in Python utilizzando 'if', 'elif', 'else'? Fornire un esempio.**

If ed elif calcolano delle condizioni che determinano l'esecuzione o meno del blocco di istruzioni ad esse associato. Ci può essere una sola if, seguita opzionalmente da 1 o più elif.

Nel caso in cui nessuna delle condizioni sia vera, il blocco else: (che è opzionale) verrà eseguito.

If `lati > 10:`

```
print('Troppi lati')
```

elif lati <= 2:

```
print('Non è un poligono')
```

else:

```
print('poligono ragionevole')
```

**DOMANDA N°13: Spiegare il concetto di alias nel contesto del passaggio di parametri ad una funzione, evidenziando se e quando si applica.**

Quando una porzione di programma chiama una funzione passando uno o più parametri, per ciascuno di questi parametri viene creato un alias, che è ricevuto dalla funzione inizializzando l'argomento relativo.

Def funz(x, y):

```
a = 5
```

```
b = 4
```

```
funz (a, b)
```

x viene inizializzato come alias ad a

y viene inizializzato come alias a b

Esempio: funz(10, a +1) : viene creato un oggetto int(10), e x viene inizializzato come riferimento a tale oggetto. Vieni calcolato a+1, e il risultato sarà il nuovo oggetto, il cui riferimento inizierà il valore di y.

**DOMANDA N°14: Spiegare almeno due approcci per confrontare numeri reali tenendo conto di possibili errori di arrotondamento.**

A, b variabili float -> valore approssimato

Quindi il confronto  $a == b$  oppure  $a != b$  non hanno senso. Allora anziché fare un controllo di uguaglianza, facciamo un controllo di uguaglianza approssimata: sono abbastanza vicine da poter essere considerate uguali.

Errore assoluto:  $\text{abs}(a-b) < \text{epsilon}$

Errore relativo:  $\text{abs}(a-b)/\text{abs}(a) < \text{epsilon}_r$

Usando la libreria standard:

```
math.isclose(a, b)
```

**DOMANDA N°15: Discutere la differenza tra un parametro formale e un parametro attuale di una funzione.**

Def funz(x, y):

```
a = 5
```

```
b = 4
```

funz (a, b)

x, y sono chiamati parametri formali. (perché sono solo dei 'nomi' interni a cui bisognerà associare un valore reale quando la funzione verrà chiamata)

a, b sono chiamati parametri attuali. (perché sono i valori effettivi con cui lavorerà la funzione).

**DOMANDA N°16: In un dizionario Python, che caratteristica fondamentale deve avere il tipo di dato usato come chiave del dizionario?**

La chiave deve essere un tipo di dato immutabile.

**DOMANDA N°17: Descrivere il fenomeno di aliasing in Python per mezzo di un esempio sulle liste. Indicare (sempre per mezzo di un esempio sulle liste) come effettuare una copia senza aliasing.**

```
a= [1, 2,3]
```

```
b= a
```

```
b.append(7)
```

allora facendo print(a) avrei [1, 2, 3, 7]

Per evitare l'aliasing potrei fare

```
B = list(a)
```

E quindi avrei a = [1, 2, 3] e b = [1, 2, 3, 7]

**DOMANDA N°18: Dire se è possibile convertire una lista in un insieme (e viceversa) e con quali effetti.**

Convertire lista in insieme?

Sì, solo se gli elementi della lista sono immutabili. I duplicati vengono eliminati e l'ordine degli elementi viene stravolto.

Convertire insieme in lista?

Si può sempre. L'ordine degli elementi è arbitrario (spesso si usa sorted() per averli in ordine noto).

**DOMANDA N°19: Qual è la differenza tra un tipo mutabile e immutabile in Python? Si elenchino un tipo mutabile ed uno immutabile a scelta.**

Un dato immutabile è un oggetto che non si può modificare dopo averlo creato/inizializzato. Si potranno solitamente creare delle nuove copie modificate, ma l'oggetto originario rimane invariato.

Un dato mutabile può essere modificato anche dopo la creazione/assegnazione iniziale.

È un concetto riferito ai valori, non alle variabili che ne fanno riferimento.

Dati immutabili: numeri interi, reali, stringhe, tuple.

Dati mutabili: liste, insiemi, dizionari, file.

**DOMANDA N°20: È possibile che in un dizionario siano presenti due chiavi associate a valori uguali? E due valori associati alla stessa chiave?**

In un dizionario ci possono essere due chiavi associate allo stesso valore?

Sì, non c'è problema, anzi non me ne accorgo nemmeno.

In un dizionario ci possono essere due valori associati alla stessa chiave?

No, le chiavi sono univoche (come un set) e a ciascuna chiave è associato un solo valore. Una chiave può essere associata ad un contenitore che, quindi, contiene valori multipli.

Es: voti['studente'] = [29, 18, 30, 23]

**DOMANDA N°21: Si supponga di leggere un dato da tastiera. Cosa succede se si usa tale dato direttamente in un'operazione aritmetica? Come si dovrebbe procedere per gestire dati numerici introdotti da tastiera?**

Dato = input()

X = dato/2

Errore perché input() restituisce sempre un dato di tipo stringa.

Se voglio fare operazioni aritmetiche dovrò prima convertirlo in int() o float(), ovviamente se ciò non genera eccezioni (es: ValueError) conviene quindi inserire la conversione in un blocco try... except.