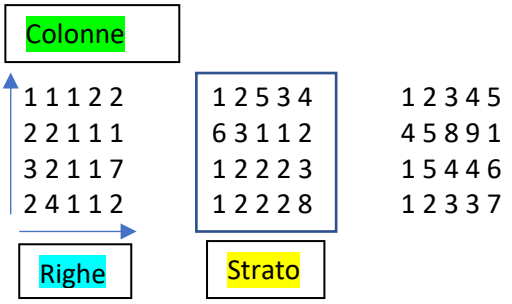
Supponendo di avere un generico array a 3 dimensioni (definito dal prof. come torta), tipo:

int X[3][4][5]

Il prof considera strati la prima dimensione, le righe sono la seconda e le colonne la terza.
A questo punto si considerano due cose: possono esserci tutti gli elementi definiti oppure solo alcuni di questi.
Per esempio avendo (su un totale di 3*4*5 elementi, quindi 60), presentiamo i due casi.



Questo è il caso in cui tutti gli elementi sono completi. Ci può essere il caso in cui non tutti siano effettivamente completi, come ad esempio (con 38 elementi):

1 1 1 2 2 1 2 5 3 4
2 2 1 1 1 6 3 1 1 2
3 2 1 1 7 1 2 2 2 3
2 4 1 1 2 1 2 2

Si vede come sia riempito totalmente il primo strato, il secondo solo parzialmente ed il terzo non completamente. In questo caso, si devono adottare alcuni accorgimenti.
Si vede l'array come ad una dimensione (ad es. *X), e si devono usare le dimensioni precedenti (strati, righe e colonne), per capire come spostarsi. La chiave è pensarli come fossero sovrapposti e usare sempre la variabile nelem o comunque quella che considera tutti gli elementi.

È uguale dire: lim1, lim2 o lim3 e dire strati, righe e colonne.
Si intende che strati=3, righe=4, colonne=5.
nhf e nvf non importa se le righe/colonne siano complete o meno

Quindi, se volessi calcolare:

- Numero di strati pieni (nsp)	nsp=nelem/righe*colonne	(qui 2)
- Numero elementi ultimo strato (eus)	eus=nelem%(righe*colonne)	(qui 18)
- Numero di righe piene (nrp)	nrp=eus/colonne	(qui 7)
- Numero di elementi restanti ultima riga (neur)	neur=eus%colonne	(qui 3)
- Lunghezza della h-fetta incompleta	lung_h=colonne*nsp	(qui 10)
- Lunghezza della v-fetta incompleta	lung_v=righe*nsp	(qui 8)
- Numero di h-fette totali	nhf=nelem/colonne	(qui 8)
- Numero di v-fette totali	nvf=nelem/righe	(qui 9)

Come conseguenza, se io mi sto spostando nelle h-fette, faccio un ciclo minore di nhf, ciclo solo sugli strati pieni (nsp) e controllo sempre la posizione della fetta, essendo sovrapposte le dimensioni.
Quindi se ho la fetta e questa è minore delle neur (righe ultimo strato), sommo le colonne (lim3) altrimenti, sommo per gli elementi ultima riga (neur sempre).
Lo stesso vale nelle v-fette, ma per ogni spostamento mi devo spostare sulle colonne sommandone il relativo indice, in questo caso semplicemente facendo (+f, dove f è l'indice della fetta).

Per approfondimenti in merito, si rimanda al mio file *Programmazione semplice (per davvero)*, dove tratto problemi di questo tipo (CTRL+F e si cerca ad esempio *pieni* e si trovano problemi corretti del tipo).
L'unico modo di vederlo è farlo in pratica.

Ora invece descriviamo un'altra visione della stessa operazione, quindi le h-fette e v-fette. Ciò significa semplicemente scansionare l'array in due modi particolari, quindi:

1 1 1 2 2	1 2 5 3 4	1 2 3 4 5
2 2 1 1 1	6 3 1 1 2	4 5 8 9 1
3 2 1 1 7	1 2 2 2 3	1 5 4 4 6
2 4 1 1 2	1 2 2 2 8	1 2 3 3 7

1 1 1 2 2	1 2 5 3 4	1 2 3 4 5
2 2 1 1 1	6 3 1 1 2	4 5 8 9 1
3 2 1 1 7	1 2 2 2 3	1 5 4 4 6
2 4 1 1 2	1 2 2 2 8	1 2 3 3 7

Ora, all'atto pratico, si vede come per spostarsi da un punto ad un altro dell'array, sia necessaria effettuare delle operazioni di divisione e moltiplicazione per prendere esattamente il punto giusto.
Si deve inoltre considerare che essi possano essere definiti o meno.
Per fare ciò si devono preventivamente calcolare quanto detto sopra.

Quindi farò:

- un ciclo esterno che scorre prima gli strati pieni
- un ciclo interno che scorre le righe piene e uso l'indice i per calcolare:
  H-fette = strati + colonne * lim2 + riga_attuale
- un ciclo interno che scorre le righe piene e uso l'indice i per calcolare:
  V-fette = strati + righe * lim3 + colonna_attuale