

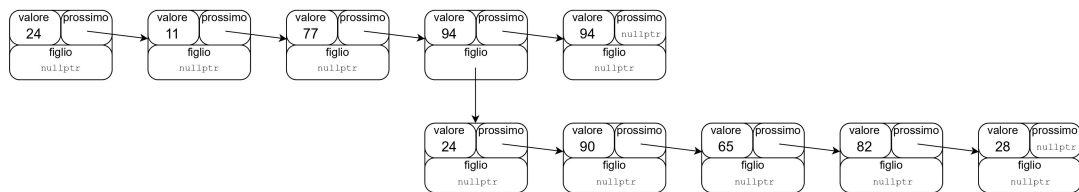
Esercizio 3

(1) Esercizio 3 v1

ESSAY marked out of 10 penalty 0 File picker

Una lista concatenata multilivello è una lista concatenata in cui ogni nodo della lista ha un valore intero e due puntatori, uno al nodo `prossimo` — come in una normale lista concatenata — e uno a un nodo `figlio`. Un nodo figlio è la testa di un'altra lista concatenata multilivello. Può esistere al massimo un nodo figlio per ogni livello.

Questo è una rappresentazione grafica di una lista concatenata multilivello con 2 livelli e 5 nodi per livello. Notare come solo un nodo per livello ha un puntatore nel campo `figlio`, mentre il campo `figlio` degli altri nodi è un `nullptr`:



Una lista concatenata multilivello può essere mappata in una matrice `m` così come segue:

- il valore del nodo i -esimo del primo livello della lista viene inserito in `m[0][i]`;
- il valore del nodo j -esimo del secondo livello della lista viene inserito in `m[1][j+i]`, dove i è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del primo livello il cui figlio è la testa della lista del secondo livello;
- generalizzando, il valore del nodo k -esimo del x -esimo livello della lista viene inserito in `m[x-1][k+h]`, dove h è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del livello $x-1$ -esimo il cui figlio è la testa della lista del x -esimo livello;
- Le celle vuote della matrice hanno valore `-1`.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `convertiListaInMatrice` che prende come parametri formali un puntatore `testa` a una struttura `Nodo` che rappresenta una lista concatenata multilivello, e due interi `righe` e `colonne` che già contengono le dimensioni della matrice. La funzione `convertiListaInMatrice` deve mappare la lista concatenata multilivello in una matrice allocata dinamicamente seguendo le regole di cui sopra. La funzione `convertiListaInMatrice` deve infine ritornare il puntatore alla matrice.

Questi sono quattro diversi esempi di esecuzione, dove il simbolo `>` rappresenta un puntatore a un nodo successivo e il simbolo `||` rappresenta un puntatore a un nodo figlio:

```

computer > ./a.out
Stampa lista multilivello
24>11>77>94>71
||
24>90>65>82>28
Stampa matrice
24 11 77 94 71 -1 -1 -1
-1 -1 -1 24 90 65 82 28
  
```

```

computer > ./a.out
Stampa lista multilivello
61>76>83>40>71
||
27>53>82>53>26
||
42>46>12>99>71
Stampa matrice
61 76 83 40 71 -1 -1 -1
27 53 82 53 26 -1 -1 -1
  
```

		-1 -1 -1 -1 42 46 12 99 71
computer > ./a.out		computer > ./a.out
Stampa lista multilivello		Stampa lista multilivello
46>29>69>73>64		70>75>37>78>33
Stampa matrice		
46 29 69 73 64		97>98>56>23>19
		Stampa matrice
		70 75 37 78 33 -1 -1 -1 -1
		-1 -1 -1 -1 97 98 56 23 19

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `convertiListaInMatrice` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`;
- Notare come la funzione `main` già contiene il codice necessario per deallocare sia la matrice che la lista concatenata multilivello. Tuttavia, ricordarsi di deallocare eventuale altra memoria allocata dinamicamente all'interno della funzione `convertiListaInMatrice`.

[esercizio3.cc](#)

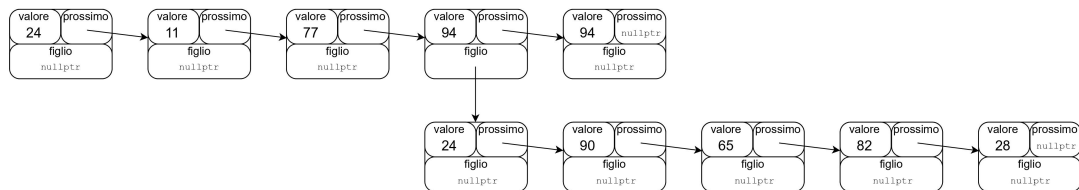
Information for graders:

(2) Esercizio 3 v2

ESSAY marked out of 10 penalty 0 File picker

Una lista concatenata multilivello è una lista concatenata in cui ogni nodo della lista ha un valore intero e due puntatori, uno al nodo `prossimo` — come in una normale lista concatenata — e uno a un nodo `figlio`. Un nodo figlio è la testa di un'altra lista concatenata multilivello. Può esistere al massimo un nodo figlio per ogni livello.

Questo è una rappresentazione grafica di una lista concatenata multilivello con 2 livelli e 5 nodi per livello. Notare come solo un nodo per livello ha un puntatore nel campo `figlio`, mentre il campo `figlio` degli altri nodi è un `nullptr`:



Una lista concatenata multilivello può essere mappata in una matrice `m` così come segue:

- il valore del nodo i -esimo del primo livello della lista viene inserito in `m[0][i]`;
- il valore del nodo j -esimo del secondo livello della lista viene inserito in `m[1][j+i]`, dove i è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del primo livello il cui figlio è la testa della lista del secondo livello;
- generalizzando, il valore del nodo k -esimo del x -esimo livello della lista viene inserito in `m[x-1][k+h]`, dove h è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del livello $x-1$ -esimo il cui figlio è la testa della lista del x -esimo livello;
- Le celle vuote della matrice hanno valore `-2`.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `convertiListaInMatrice` che prende come parametri formali un puntatore `testa` a una struttura `Nodo` che rappresenta una lista concatenata multilivello, e due interi `righe` e `colonne` che già contengono le dimensioni della matrice. La funzione `convertiListaInMatrice` deve mappare la lista concatenata multilivello in una matrice allocata dinamicamente seguendo le regole di cui sopra. La funzione `convertiListaInMatrice` deve infine ritornare il puntatore alla matrice.

Questi sono quattro diversi esempi di esecuzione, dove il simbolo `>` rappresenta un puntatore a un nodo successivo e il simbolo `||` rappresenta un puntatore a un nodo figlio:

<pre>computer > ./a.out Stampa lista multilivello 24>11>77>94>71 24>90>65>82>28 Stampa matrice 24 11 77 94 71 -2 -2 -2 -2 -2 -2 24 90 65 82 28</pre>		<pre>computer > ./a.out Stampa lista multilivello 61>76>83>40>71 27>53>82>53>26 42>46>12>99>71 Stampa matrice 61 76 83 40 71 -2 -2 -2 -2 27 53 82 53 26 -2 -2 -2 -2 -2 -2 -2 -2 42 46 12 99 71</pre>
<pre>computer > ./a.out Stampa lista multilivello 46>29>69>73>64</pre>		<pre>computer > ./a.out Stampa lista multilivello 70>75>37>78>33</pre>

Stampa matrice		
46 29 69 73 64		97>98>56>23>19
		Stampa matrice
		70 75 37 78 33 -2 -2 -2 -2
		-2 -2 -2 -2 97 98 56 23 19

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `convertiListaInMatrice` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`;
- Notare come la funzione `main` già contiene il codice necessario per deallocare sia la matrice che la lista concatenata multilivello. Tuttavia, ricordarsi di deallocare eventuale altra memoria allocata dinamicamente all'interno della funzione `convertiListaInMatrice`.

[esercizio3.cc](#)

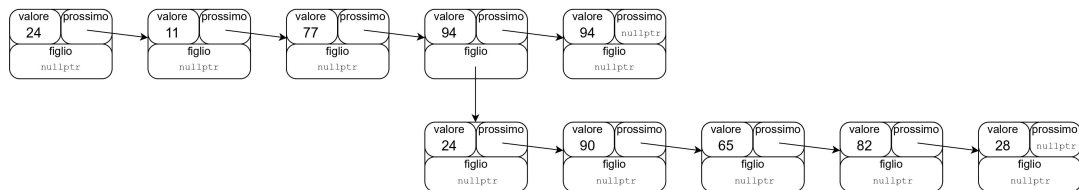
Information for graders:

(3) Esercizio 3 v3

ESSAY marked out of 10 penalty 0 File picker

Una lista concatenata multilivello è una lista concatenata in cui ogni nodo della lista ha un valore intero e due puntatori, uno al nodo `prossimo` — come in una normale lista concatenata — e uno a un nodo `figlio`. Un nodo figlio è la testa di un'altra lista concatenata multilivello. Può esistere al massimo un nodo figlio per ogni livello.

Questo è una rappresentazione grafica di una lista concatenata multilivello con 2 livelli e 5 nodi per livello. Notare come solo un nodo per livello ha un puntatore nel campo `figlio`, mentre il campo `figlio` degli altri nodi è un `nullptr`:



Una lista concatenata multilivello può essere mappata in una matrice `m` così come segue:

- il valore del nodo i -esimo del primo livello della lista viene inserito in `m[0][i]`;
- il valore del nodo j -esimo del secondo livello della lista viene inserito in `m[1][j+i]`, dove i è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del primo livello il cui figlio è la testa della lista del secondo livello;
- generalizzando, il valore del nodo k -esimo del x -esimo livello della lista viene inserito in `m[x-1][k+h]`, dove h è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del livello $x-1$ -esimo il cui figlio è la testa della lista del x -esimo livello;
- Le celle vuote della matrice hanno valore 10.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `convertiListaInMatrice` che prende come parametri formali un puntatore `testa` a una struttura `Nodo` che rappresenta una lista concatenata multilivello, e due interi `righe` e `colonne` che già contengono le dimensioni della matrice. La funzione `convertiListaInMatrice` deve mappare la lista concatenata multilivello in una matrice allocata dinamicamente seguendo le regole di cui sopra. La funzione `convertiListaInMatrice` deve infine ritornare il puntatore alla matrice.

Questi sono quattro diversi esempi di esecuzione, dove il simbolo `>` rappresenta un puntatore a un nodo successivo e il simbolo `||` rappresenta un puntatore a un nodo figlio:

```
computer > ./a.out
Stampa lista multilivello
24>11>77>94>71
||
24>90>65>82>28
Stampa matrice
24 11 77 94 71 10 10 10
10 10 10 24 90 65 82 28

computer > ./a.out
Stampa lista multilivello
46>29>69>73>64

computer > ./a.out
Stampa lista multilivello
61>76>83>40>71
||
27>53>82>53>26
||
42>46>12>99>71
Stampa matrice
61 76 83 40 71 10 10 10
27 53 82 53 26 10 10 10
10 10 10 10 42 46 12 99 71

computer > ./a.out
Stampa lista multilivello
70>75>37>78>33
```

Stampa matrice		
46 29 69 73 64		97>98>56>23>19
		Stampa matrice
		70 75 37 78 33 10 10 10 10
		10 10 10 10 97 98 56 23 19

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `convertiListaInMatrice` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`;
- Notare come la funzione `main` già contiene il codice necessario per deallocare sia la matrice che la lista concatenata multilivello. Tuttavia, ricordarsi di deallocare eventuale altra memoria allocata dinamicamente all'interno della funzione `convertiListaInMatrice`.

[esercizio3.cc](#)

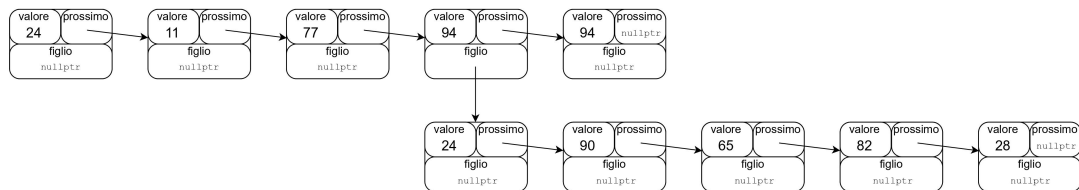
Information for graders:

(4) Esercizio 3 v4

ESSAY marked out of 10 penalty 0 File picker

Una lista concatenata multilivello è una lista concatenata in cui ogni nodo della lista ha un valore intero e due puntatori, uno al nodo `prossimo` — come in una normale lista concatenata — e uno a un nodo `figlio`. Un nodo figlio è la testa di un'altra lista concatenata multilivello. Può esistere al massimo un nodo figlio per ogni livello.

Questo è una rappresentazione grafica di una lista concatenata multilivello con 2 livelli e 5 nodi per livello. Notare come solo un nodo per livello ha un puntatore nel campo `figlio`, mentre il campo `figlio` degli altri nodi è un `nullptr`:



Una lista concatenata multilivello può essere mappata in una matrice `m` così come segue:

- il valore del nodo i -esimo del primo livello della lista viene inserito in `m[0][i]`;
- il valore del nodo j -esimo del secondo livello della lista viene inserito in `m[1][j+i]`, dove i è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del primo livello il cui figlio è la testa della lista del secondo livello;
- generalizzando, il valore del nodo k -esimo del x -esimo livello della lista viene inserito in `m[x-1][k+h]`, dove h è l'indice della riga nella matrice in cui è stato inserito il valore del nodo del livello $x-1$ -esimo il cui figlio è la testa della lista del x -esimo livello;
- Le celle vuote della matrice hanno valore 20.

Scrivere nel file `esercizio3.cc` la corretta implementazione della funzione `convertiListaInMatrice` che prende come parametri formali un puntatore `testa` a una struttura `Nodo` che rappresenta una lista concatenata multilivello, e due interi `righe` e `colonne` che già contengono le dimensioni della matrice. La funzione `convertiListaInMatrice` deve mappare la lista concatenata multilivello in una matrice allocata dinamicamente seguendo le regole di cui sopra. La funzione `convertiListaInMatrice` deve infine ritornare il puntatore alla matrice.

Questi sono quattro diversi esempi di esecuzione, dove il simbolo `>` rappresenta un puntatore a un nodo successivo e il simbolo `||` rappresenta un puntatore a un nodo figlio:

<pre>computer > ./a.out Stampa lista multilivello 24>11>77>94>71 24>90>65>82>28 Stampa matrice 24 11 77 94 71 20 20 20 20 20 20 24 90 65 82 28</pre>		<pre>computer > ./a.out Stampa lista multilivello 61>76>83>40>71 27>53>82>53>26 42>46>12>99>71 Stampa matrice 61 76 83 40 71 20 20 20 27 53 82 53 26 20 20 20 20 20 20 20 42 46 12 99 71</pre>
<pre>computer > ./a.out Stampa lista multilivello 46>29>69>73>64</pre>		<pre>computer > ./a.out Stampa lista multilivello 70>75>37>78>33</pre>

Stampa matrice		
46 29 69 73 64		97>98>56>23>19
		Stampa matrice
		70 75 37 78 33 20 20 20 20
		20 20 20 20 97 98 56 23 19

Note:

- Scaricare il file `esercizio3.cc`, modificarlo per inserire la corretta implementazione della funzione `convertiListaInMatrice` e infine caricare il file risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito;
- Ricordarsi di distinguere gli esempi nella descrizione dell'esercizio (che servono solo ad aiutare a comprendere il problema) dalle istruzioni di implementazione;
- E' consentito definire ed implementare funzioni ausiliarie che possano aiutarvi nella soluzione del problema;
- All'interno di questo programma non e' ammesso l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`;
- Notare come la funzione `main` già contiene il codice necessario per deallocare sia la matrice che la lista concatenata multilivello. Tuttavia, ricordarsi di deallocare eventuale altra memoria allocata dinamicamente all'interno della funzione `convertiListaInMatrice`.

[esercizio3.cc](#)

Information for graders:

Total of marks: 40