

Il Triangolo di Tartaglia contiene molte informazioni interessanti. Implementare il tipo di dato astratto Tartaglia, dotandolo delle seguenti funzionalità:

--- Metodi invocati nella PRIMA PARTE di main.cpp: ---

✓ Tartaglia t(n);

Costruttore, che crea un nuovo oggetto Tartaglia di ordine n. Un oggetto di ordine n è composto da (n+1) righe, in cui nella riga di indice i si trovano i coefficienti di  $(a+b)^i$ , per  $i = 0, 1, \dots, n$ .

Tutte le righe di un triangolo di Tartaglia iniziano e terminano con un uno. Inoltre il coefficiente nella riga i, colonna j (con j diverso dalla prima e dall'ultima colonna valida) si ottiene sommando i due elementi adiacenti della riga precedente, in posizione rispettivamente j-1 e j, come illustrato nella figura sottostante.

	0	1	2	3	4	5	6
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1

✓ cout << t;

Operatore di uscita per oggetti di tipo triangolo di Tartaglia. Un oggetto triangolo di Tartaglia deve essere visualizzato nel seguente modo (nella figura è mostrato un triangolo di ordine 9):

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

```

```

____1
____1____1          (versione con spazi in evidenza)
____1____2____1
____1____3____3____1
____1____4____6____4____1
____1____5____10____10____5____1
____1____6____15____20____15____6____1
____1____7____21____35____35____21____7____1
____1____8____28____56____70____56____28____8____1
____1____9____36____84____126____126____84____36____9____1

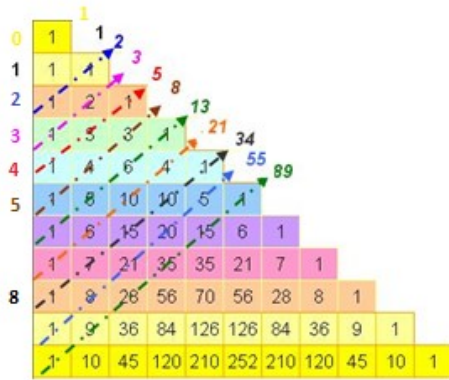
```

Si noti che l'ultimo numero della seconda colonna coincide con l'ordine nel triangolo. Si noti inoltre come i numeri siano distanziati da 4 caratteri bianchi qualora il numero successivo abbia **una sola cifra**, di 3 qualora abbia **due cifre**, di 2 qualora abbia **tre cifre** e di 1 spazio bianco qualora abbia **quattro cifre**. Così facendo vengono mostrati correttamente triangoli di Tartaglia di grado minore o uguale a 15.

**Suggerimento:** per ottenere i numeri allineati a destra come sopra e distanziati dal numero di spazi bianchi corretto, si consiglia di utilizzare la funzione `setw(5)` (`cout<<setw(5)<<...`). Per utilizzare tale funzione, occorre includere la libreria `iomanip`, nel solito modo: `#include<iomanip>`

✓ **t.fibonacci(n);**

Questa funzione restituisce l'n-esimo elemento della successione di Fibonacci, ricavandolo dal Triangolo di Tartaglia, secondo l'algoritmo illustrato nella seguente figura (**e non secondo un algoritmo diverso**):



Qualora il parametro n sia maggiore dell'ordine del triangolo o minore di 0, **la funzione deve restituire -1**.

Esempi:

```
cout<<t.fibonacci(0); // deve restituire 1
cout<<t.fibonacci(1); // deve restituire 1
cout<<t.fibonacci(2); // deve restituire 2
cout<<t.fibonacci(5); // deve restituire 8
cout<<t.fibonacci(8); // deve restituire 34
```

--- **Metodi invocati nella SECONDA PARTE di main.cpp:** ---

✓ **t.espandi(n);**

Funzione che prende in ingresso un naturale n e **stampa a video**  $(a+b)^n$ . Ad esempio:

- per n = 0 deve stampare {1}
- per n = 1 deve stampare {a^1 + b^1}
- per n = 2 deve stampare {a^2 + 2a^1\*b^1 + b^2}
- per n = 3 deve stampare {a^3 + 3a^2\*b^1 + 3a^1\*b^2 + b^3}
- per n = 4 deve stampare {a^4 + 4a^3\*b^1 + 6a^2\*b^2 + 4a^1\*b^3 + b^4}
- (ecc...)

Ovviamente i coefficienti corretti da stampare si trovano nella riga opportuna del triangolo di Tartaglia.

Se n è maggiore dell'ordine del triangolo t, la funzione **non deve stampare a video nulla**.

✓ **t2 = t;**

Operatore di assegnamento per la classe Tartaglia, che rende t2 uguale a t.

✓ **~Tartaglia();**

Distruttore per oggetti di tipo triangolo di Tartaglia.

Mediante il linguaggio C++, realizzare il tipo di dato astratto definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo string, il tipo vector, il tipo list, ecc.

```

// file main.cpp
#include "compito.h"

int main() {

    cout<<endl<<"--- PRIMA PARTE ---" << endl;

    Tartaglia t(9);
    cout<<t<<endl<<endl;

    cout<<"Test della funzione fibonacci:"<<endl;
    cout<< t.fibonacci(0)<<endl; // deve stampare 1
    cout<< t.fibonacci(1)<<endl; // deve stampare 1
    cout<< t.fibonacci(2)<<endl; // deve stampare 2
    cout<< t.fibonacci(5)<<endl; // deve stampare 8
    cout<< t.fibonacci(8)<<endl; // deve stampare 34
    cout << endl;

    cout<<endl<<"--- SECONDA PARTE ---" << endl;

    cout<<"Test della espandi (deve stampare {a^4 + 4a^3*b^1 + 6a^2*b^2 + 4a^1*b^3 + b^4})"<<endl;
    t.espandi(4);

    cout << endl<<endl;
    cout << "Creazione dell'oggetto t2 di ordine 13 e sua stampa" <<endl;
    Tartaglia t2(13);
    cout<<t2<<endl<<endl;

    cout << "Test dell'operatore di assegnamento" <<endl;
    t2 = t;
    cout<<t2<<endl;

}

```

**USCITA CHE DEVE PRODURRE IL PROGRAMMA**

```

--- PRIMA PARTE ---
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

Test della funzione fibonacci:
1
1
2
8
34

--- SECONDA PARTE ---
Test della espandi (deve stampare {a^4 + 4a^3*b^1 + 6a^2*b^2 + 4a^1*b^3 + b^4})
{a^4 + 4a^3*b^1 + 6a^2*b^2 + 4a^1*b^3 + b^4}

Creazione dell'oggetto t2 di ordine 13 e sua stampa
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1

Test dell'operatore di assegnamento
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

```