Il tipo di dato astratto `Distributore` implementa le funzionalità di un distributore automatico di vivande. Un distributore è composto da `N` ripiani, numerati a partire da 1. Per semplicità, si assuma che `N` non possa essere maggiore di 9. Ogni ripiano ha 5 slot per contenere i pezzi in vendita. Gli slot sono numerati a partire da 0. I pezzi di uno slot sono tutti dello stesso tipo e ogni slot può contenere fino a un massimo di 5 pezzi. L'utente può acquistare un pezzo alla volta, selezionando il numero di uno slot, che è un numero intero a due cifre XY, dove X identifica il ripiano e Y rappresenta lo slot di tale ripiano (per esempio, 23 rappresenta lo slot numero 3 del secondo ripiano). Implementare le seguenti operazioni che possono essere fatte su `Distributore`:

--- PRIMA PARTE --- *(qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte,* ***l'intera prova sarà considerata insufficiente*** *e pertanto* ***non sarà corretta****)*

✓ **`Distributore d(N);`**

Costruttore che inizializza un distributore avente `N` ripiani. Inizialmente, tutti gli slot contengono un pezzo.

✓ **`d.acquista(i);`**

Operazione che implementa l'acquisto di un pezzo dallo slot avente numero `i`. La funzione decrementa il numero di pezzi disponibili in tale slot. Se lo slot non contiene pezzi, la funzione restituisce `false`, altrimenti la funzione restituisce `true`.

✓ **`d.aggiungi(i,n);`**

Operazione che aggiunge `n` pezzi nello slot avente numero `i`. Se lo slot non ha abbastanza posti disponibili, aggiunge solo i pezzi possibili fino al raggiungimento del numero massimo.

`cout << d;`

Operatore di uscita per il tipo `Distributore`. L'uscita ha il seguente formato:

```
1: 1 3 0 1 1
2: 1 1 5 1 0
3: 0 4 1 1 1
```

L'output mostrato corrisponde a un distributore avente 3 ripiani, dove per ogni slot viene mostrato il numero di pezzi contenuti.

--- SECONDA PARTE ---

✓ **`Distributore d1(d);`**

Costruttore di copia per il tipo Distributore, che crea un distributore `d1` uguale a `d`.

✓ **`d + n;`**

Operatore che restituisce un nuovo distributore uguale a d ma con con l'aggiunta di n ripiani. I ripiani sono aggiunti con il più alto numero d'ordine e tutti gli slot sono vuoti.
Esempio: effettuando l'operazione `d+2` sull'esempio precedente, si ottiene il seguente distributore, dove sono stati aggiunti 2 ripiani con tutti slot vuoti:

```
1: 1 3 0 1 1
2: 1 1 5 1 0
3: 0 4 1 1 1
4: 0 0 0 0 0
5: 0 0 0 0 0
```

✓ **`~Distributore();`**

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto **`Distributore`**, definito dalle precedenti specifiche. **Gestire le eventuali situazioni di errore**.

USCITA CHE DEVE PRODURRE IL PROGRAMMA

```
--- PRIMA PARTE ---
Test costruttore
1: 1 1 1 1 1
2: 1 1 1 1 1
3: 1 1 1 1 1

Test acquista
1: 1 1 0 1 1
2: 1 1 1 1 0
3: 0 1 1 1 1

Test aggiungi
1: 1 3 0 1 1
2: 1 1 5 1 0
3: 0 4 1 1 1

--- SECONDA PARTE ---
Test operator+
1: 1 3 0 1 1
2: 1 1 5 1 0
3: 0 4 1 1 1
4: 0 0 0 0 0
5: 0 0 0 0 0

Test costr. di copia
1: 1 3 0 1 1
2: 1 1 5 1 0
3: 0 4 1 1 1

(d1 e' stato distrutto)
```