

Gestione di una **Lista concatenata e ordinata** di numeri **allocata dinamicamente** mediante **puntatori**

Il programma *listaNUM.ccp* gestisce una **lista concatenata** di numeri **allocati dinamicamente** utilizzando **puntatori**. La lista viene **ordinata** per valori crescenti già dal momento dell'inserimento dei valori in lista.

Una lista gestita mediante puntatori viene costruita definendo una struttura dei questo tipo:

```
struct Nodo {
    int n;
    Nodo* pSucc;
};
Nodo* pTesta = NULL;
Nodo* pNew = NULL;
```

Non vengono allocate variabili al caricamento dell' eseguibile in Memoria Centrale, ma solo i puntatori **pTesta** e **pNew**.

Solo a seguito di richieste di inserimento di un numero viene allocata una istanza della struttura Nodo con l'istruzione **pNew = new Nodo**; che salva l'indirizzo del nuovo nodo in **pNew** e poi nella catena di puntatori.

Nella ipotesi che la lista sia già piena e contenga 4 nodi ordinati per i valori 3, 7, 9 e 11, lo **scorrimento della lista** può essere realizzato come segue:

Istruzione	Effetto	Descrizione
<pre>Nodo* pNext; pNext = pTesta;</pre>		<p>pNext assume il valore di pTesta, quindi vale NULL (se la lista è vuota) oppure punta al primo Nodo della lista (il Nodo 3 nell'esempio)</p>
<pre>while (pNext != NULL) { cout << pNext -> n << ' '; pNext = pNext -> pSucc; }</pre>	<p>fin quando pNext punta a un Nodo (ie, non è stata raggiunta la fine della lista) viene esposta l'informazione contenuta nel Nodo, cioè pNext -> n e aggiornato il puntatore pNext al prossimo nodo puntato da pNext -> pSucc</p>	<p>raggiunto e visualizzato l'ultimo Nodo (11), l'ultimo aggiornamento di pNext lo fissa a NULL interrompendo il ciclo <i>while</i></p>

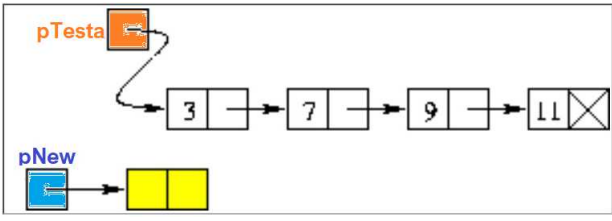
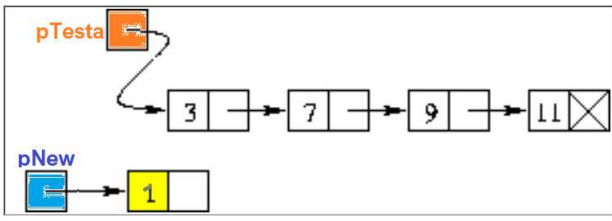
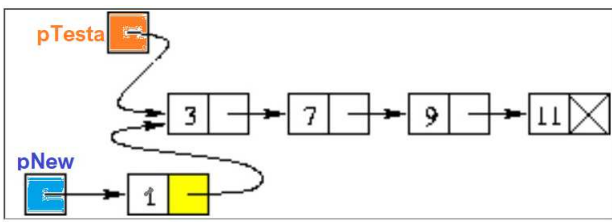
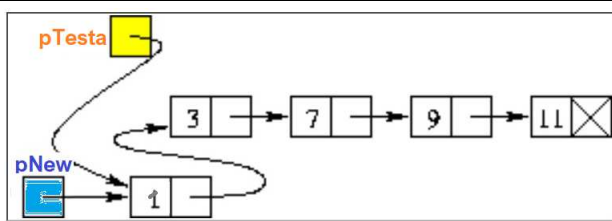
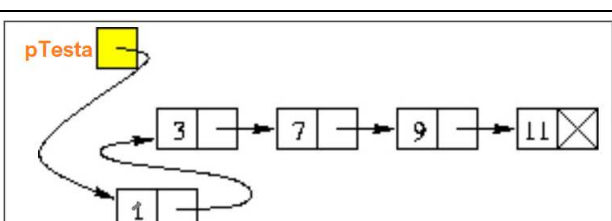
NOTA: il indica il valore NULL per il puntatore **pSucc** dell'ultimo nodo

Alla richiesta di **inserimento di un nuovo numero** va creato un nuovo Nodo (allocato in M.C.) e per, l'**inserimento nella lista** ordinata, va individuato il primo elemento della lista con n maggiore del nuovo valore da inserire e vanno aggiornati i puntatori del nuovo elemento creato e dell'elemento che deve precederlo nella lista. L'inserimento prevede azioni diverse per i seguenti casi:

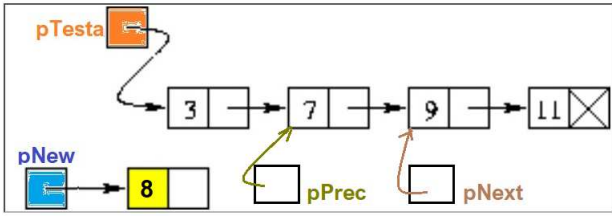
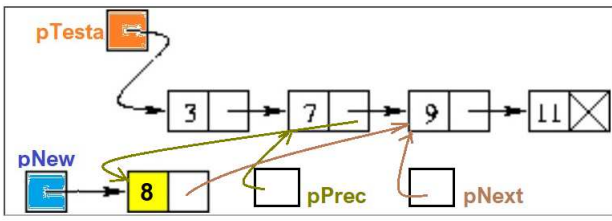
- A. Lista vuota (è sufficiente far puntare **pTesta** all'indirizzo del Nuovo nodo inserito **pNew**)
- B. Nodo da inserire come primo della lista (n minore del primo elemento)
- C. Nodo da inserire come ultimo della lista (n maggiore dell'ultimo)
- D. Nodo da inserire in una posizione intermedia

Gestione di una **Lista concatenata e ordinata** di numeri **allocata dinamicamente** mediante **puntatori**

Caso B. Si vuole inserire il valore **1**, questo valore (il Nodo che lo contiene) va collocato nella lista prima di **3**, cioè come primo elemento puntato da **pTesta**; questi i passi da eseguire:

Istruzione	Effetto	Descrizione
<code>pNew = new Nodo;</code>		Creazione del nuovo Nodo (in giallo) con l'aiuto del puntatore di appoggio pNew
<code>pNew ->n = 1;</code>		Scrittura del numero da inserire (esempio 1) nel nuovo Nodo allocato in memoria e il cui indirizzo è salvato in pNew
<code>pNew->pSucc = pTesta;</code>		Impostazione del puntatore del nuovo Nodo (pNew->pSucc) per puntare al primo elemento della lista il cui indirizzo è attualmente salvato in pTesta
<code>pTesta = pNew;</code>		Aggiornamento di pTesta (puntatore al primo Nodo) con l'indirizzo contenuto in pNew per puntare al nuovo Nodo creato
		A questo punto il puntatore pNew può essere riutilizzato per allocare nuovi Nodi

Caso D. Raggiunto con **pNext** il primo Nodo con valore maggiore del nuovo valore 8 (nell'esempio 9)

<code>pPrec = pNext;</code> <code>pNext = pNext -> pSucc;</code>		Avendo salvato nel ciclo di lettura l'indirizzo del nodo precedente (nell'esempio 7) in pPrec prima di far avanzare pNext
<code>pNew -> pSucc = pNext;</code> <code>pPrec -> pSucc = pNew;</code>		Si procede ad aggiornare - il puntatore del nuovo nodo (pNew ->pSucc) con pNext - il puntatore del nodo prec. (pPrec -> pSucc) con pNew