

# Progetto e processo di sviluppo

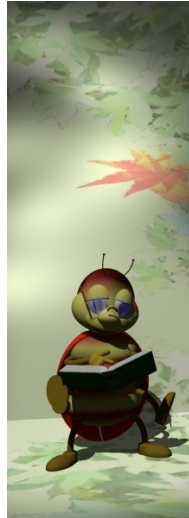
Claudio Mirolo

Dipartimento di Matematica e Informatica  
Università di Udine

PLS 2013–14

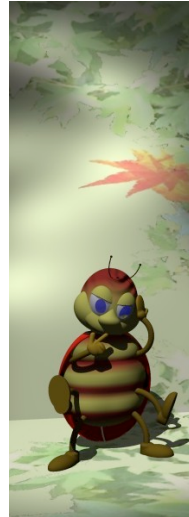
# Sommario

- 1 **Retrospettiva**
  - osservazioni da sfruttare
  - nei panni di un ingegnere
  
- 2 **Artefatto**
  - progetto
  - processo di sviluppo
  
- 3 **Epilogo**
  - ad libitum

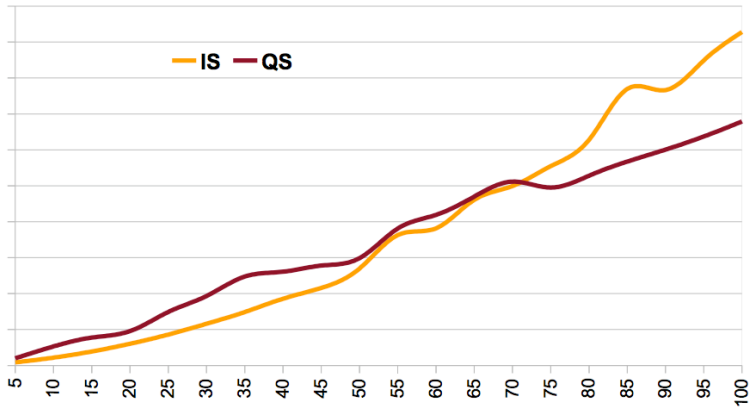


# Trama

- 1 **Retrospettiva**
  - osservazioni da sfruttare
  - nei panni di un ingegnere
- 2 **Artefatto**
  - progetto
  - processo di sviluppo
- 3 **Epilogo**
  - ad libitum



# Osservazioni: inversione delle prestazioni





# L'informatico come "artigiano"

Uno scienziato costruisce per studiare;  
un ingegnere studia per costruire.

F.P. Brooks, Jr. 1996

# Domande da ingegnere

- C'è modo di *applicare* le conoscenze acquisite?
- È possibile migliorare le prestazioni combinando assieme i “blocchi funzionali” *QuickSort* e *InsertionSort* ?
- In quali condizioni le prestazioni di ciascuna tecnica sono più o meno vantaggiose? Quali *tradeoff* ?
- Come si può assicurare l'*affidabilità* del sistema che ne risulta? come organizzare il processo di sviluppo?
- Quali sono le condizioni “ottimali” di integrazione delle due componenti? come regolarle su base empirica?



## Domande da ingegnere

- C'è modo di *applicare* le conoscenze acquisite?
- È possibile migliorare le prestazioni combinando assieme i “blocchi funzionali” *QuickSort* e *InsertionSort* ?
- In quali condizioni le prestazioni di ciascuna tecnica sono più o meno vantaggiose? Quali *tradeoff* ?
- Come si può assicurare l'*affidabilità* del sistema che ne risulta? come organizzare il processo di sviluppo?
- Quali sono le condizioni “ottimali” di integrazione delle due componenti? come regolarle su base empirica?



## Domande da ingegnere

- C'è modo di *applicare* le conoscenze acquisite?
- È possibile migliorare le prestazioni combinando assieme i “blocchi funzionali” *QuickSort* e *InsertionSort* ?
- In quali condizioni le prestazioni di ciascuna tecnica sono più o meno vantaggiose? Quali *tradeoff* ?
- Come si può assicurare l'*affidabilità* del sistema che ne risulta? come organizzare il processo di sviluppo?
- Quali sono le condizioni “ottimali” di integrazione delle due componenti? come regolarle su base empirica?





## Domande da ingegnere

- C'è modo di *applicare* le conoscenze acquisite?
- È possibile migliorare le prestazioni combinando assieme i “blocchi funzionali” *QuickSort* e *InsertionSort* ?
- In quali condizioni le prestazioni di ciascuna tecnica sono più o meno vantaggiose? Quali *tradeoff* ?
- Come si può assicurare l'*affidabilità* del sistema che ne risulta? come organizzare il processo di sviluppo?
- Quali sono le condizioni “ottimali” di integrazione delle due componenti? come regolarle su base empirica?

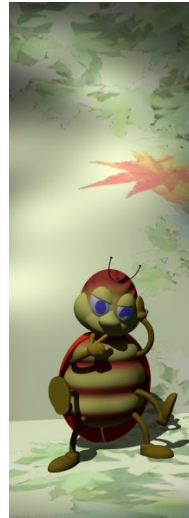


## Domande da ingegnere

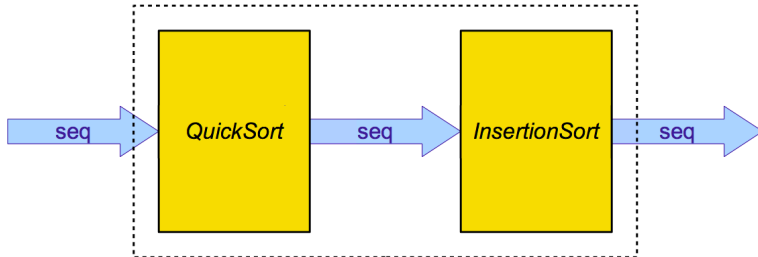
- C'è modo di *applicare* le conoscenze acquisite?
- È possibile migliorare le prestazioni combinando assieme i “blocchi funzionali” *QuickSort* e *InsertionSort* ?
- In quali condizioni le prestazioni di ciascuna tecnica sono più o meno vantaggiose? Quali *tradeoff* ?
- Come si può assicurare l'*affidabilità* del sistema che ne risulta? come organizzare il processo di sviluppo?
- Quali sono le condizioni “ottimali” di integrazione delle due componenti? come regolarle su base empirica?

# Trama

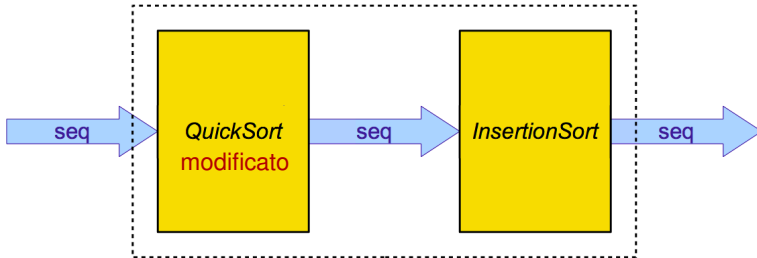
- 1 Retrospettiva
  - osservazioni da sfruttare
  - nei panni di un ingegnere
- 2 Artefatto
  - progetto
  - processo di sviluppo
- 3 Epilogo
  - ad libitum



# Progetto



# Progetto





# QuickSort modificato

```
private void quickSort( int l, int u, int[] seq ) {  
  
    if ( l + delta < u ) { // soglia  
        int m = seq[l], i = l, j = u;  
        do {  
            while ( seq[i] < m ) { i = i + 1; }  
            while ( seq[j] > m ) { j = j - 1; }  
            if ( i < j ) {  
                int x = seq[i]; seq[i] = seq[j]; seq[j] = x;  
                i = i + 1; j = j - 1;  
            }  
        } while ( i < j );  
        if ( seq[j] > m ) { j = j - 1; }  
        quickSort( l, j, seq );  
        quickSort( j+1, u, seq );  
    }  
}
```



# Assemblaggio dei blocchi

```
private final SortingAlgorithm qs, is;

public FasterSort() {

    qs = new QuickSort();           // modificato
    is = new InsertionSort();
}

public void sort( int[] seq ) {

    qs.sort( seq );
    is.sort( seq );
}
```



# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?





# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?



# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?



# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?



# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?



# Relazione struttura/funzione

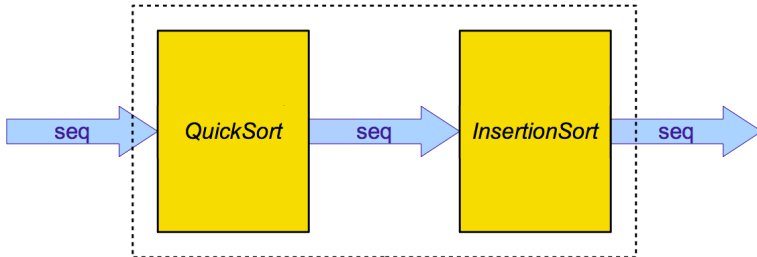
- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?



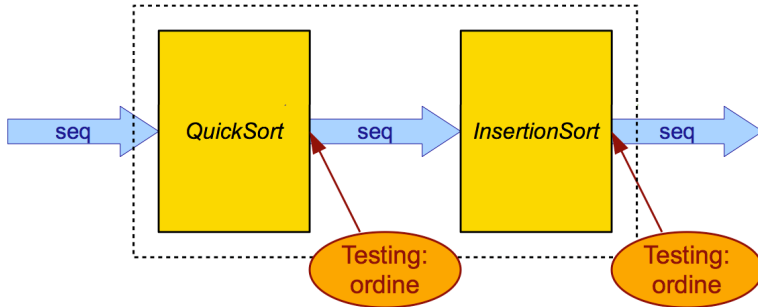
# Relazione struttura/funzione

- Dalla relazione struttura/funzione di ciascuna delle parti  
Alla relazione struttura/funzione dell'artefatto creato
- Relazione struttura/funzione ai fini dell'ordinamento
- Relazione struttura/funzione ai fini delle prestazioni
- Come cambia la funzione di *QuickSort* modificato  
al variare della soglia *delta*?  
Ai fini dell'ordinamento? Ai fini delle prestazioni?
- Quali proprietà della struttura progettata sono  
*funzionali* ad ottenere migliori prestazioni?

# Pianificazione del lavoro

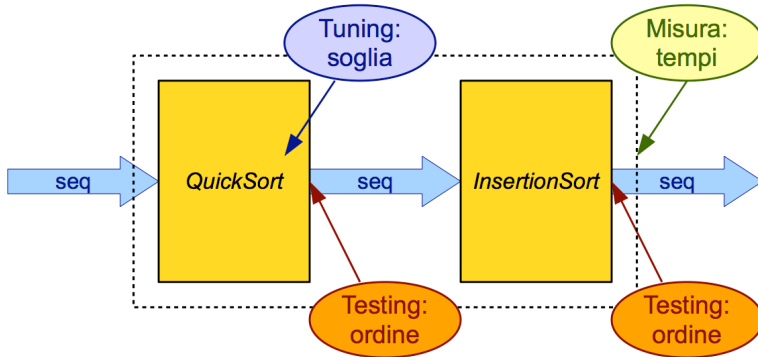


# Pianificazione del lavoro

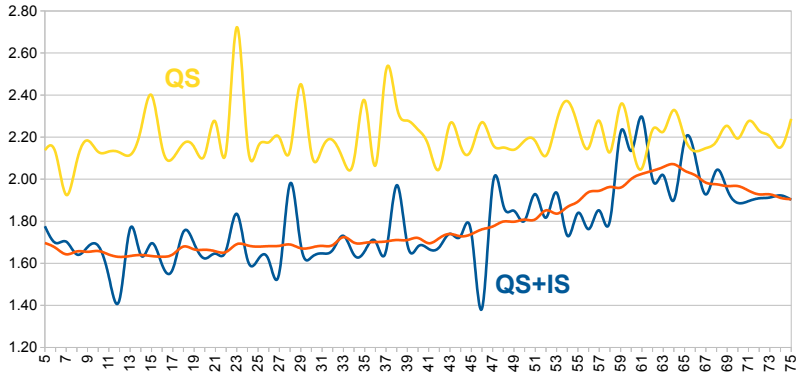




# Pianificazione del lavoro



# Regolazione empirica della soglia





# Al lavoro...



UnitTesting.java



ProjectDevelopment.java



# Al lavoro...



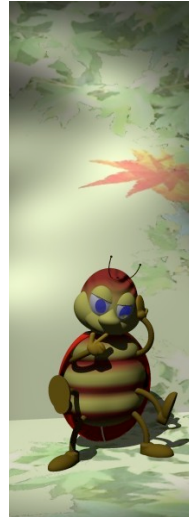
UnitTesting.java



ProjectDevelopment.java

# Trama

- 1 Retrospettiva
  - osservazioni da sfruttare
  - nei panni di un ingegnere
- 2 Artefatto
  - progetto
  - processo di sviluppo
- 3 Epilogo
  - ad libitum





# Creatività

- il progetto di un artefatto è un processo *creativo* per il quale non esistono ricette
- Specificata la *funzione* dell'artefatto, molteplici *strutture* la possono realizzare

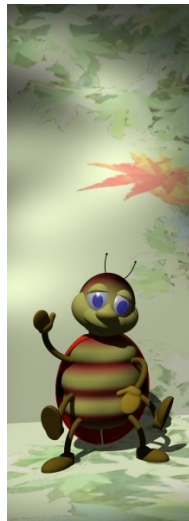


# Creatività

- il progetto di un artefatto è un processo *creativo* per il quale non esistono ricette
  
- Specificata la *funzione* dell'artefatto, molteplici *strutture* la possono realizzare

# The End

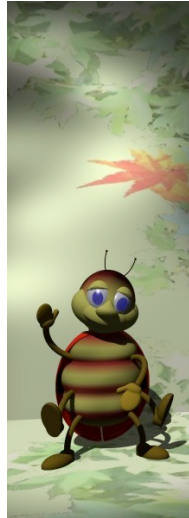
**Vostre domande?**





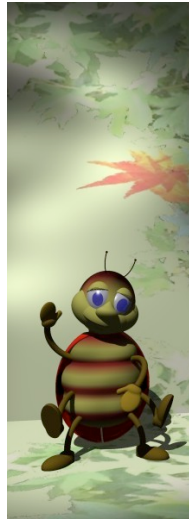
# The End

**Grazie**



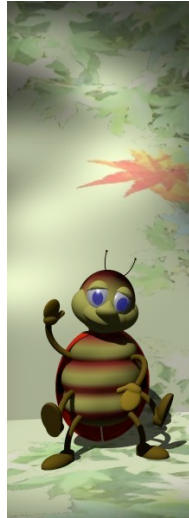
# The End

**Grazie**



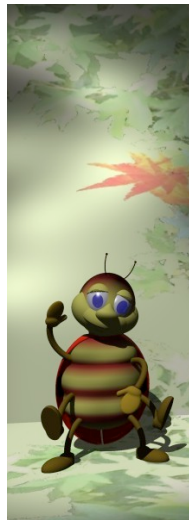
# The End

**Grazie**



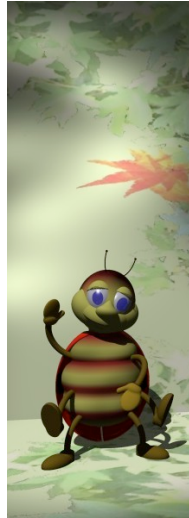
# The End

**Grazie**



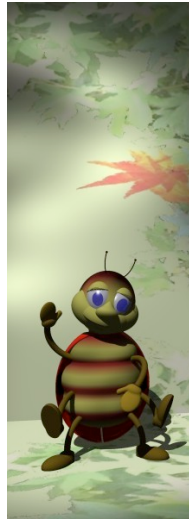
# The End

**Grazie**



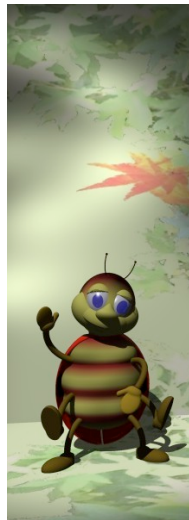
# The End

**Grazie**



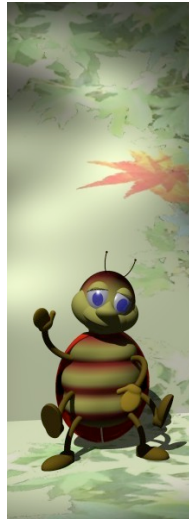
# The End

**Grazie**



# The End

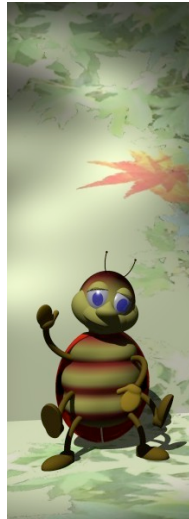
**Grazie**





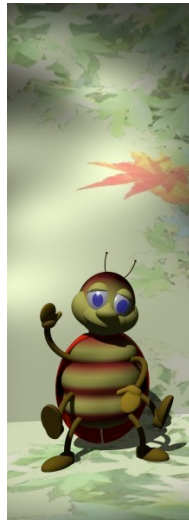
# The End

**Grazie**



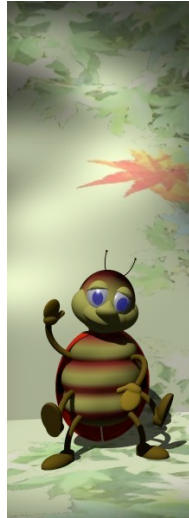
# The End

**Grazie**



# The End

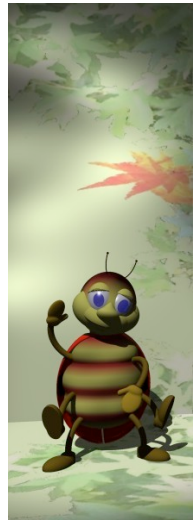
**Grazie**





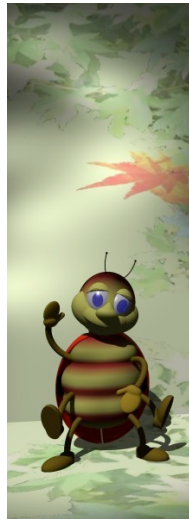
# The End

**Grazie**



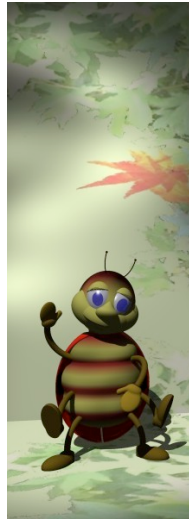
# The End

**Grazie**



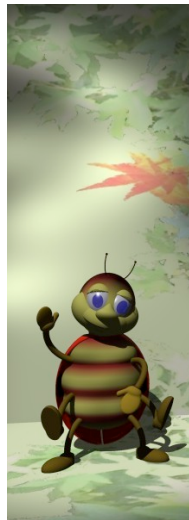
# The End

**Grazie**



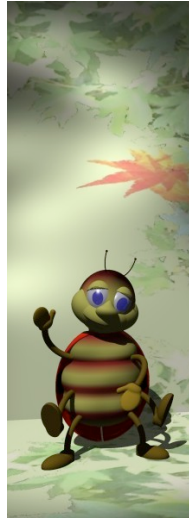
# The End

**Grazie**



# The End

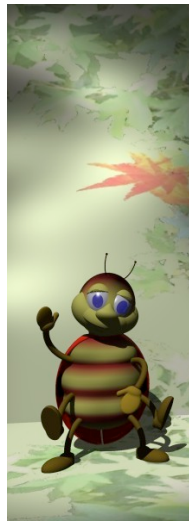
**Grazie**





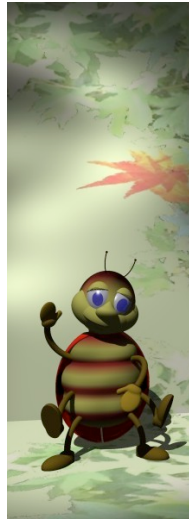
# The End

**Grazie**



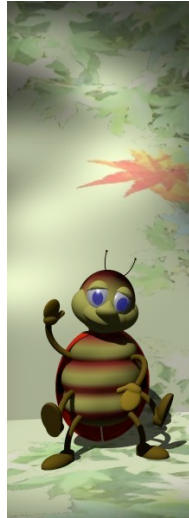
# The End

**Grazie**



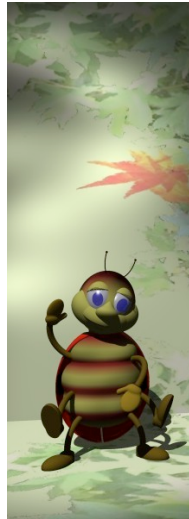
# The End

**Grazie**



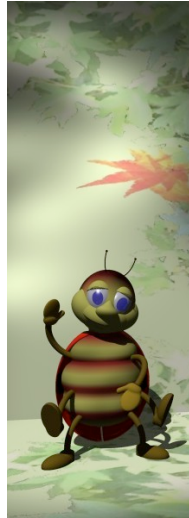
# The End

**Grazie**



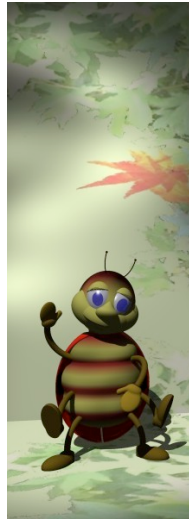
# The End

**Grazie**



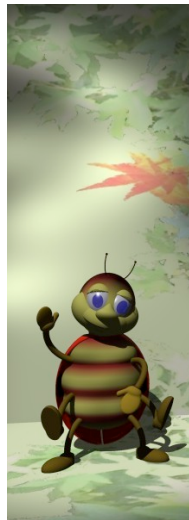
# The End

**Grazie**



# The End

**Grazie**



# The End

**Grazie**

