

# ALGORITMI E STRUTTURE DATI

GIORGIO D'ANGELO

## 1) Hashtable di disjoint set

Si supponga di dover memorizzare un insieme di coppie <key,value> contenute in un file. key è numero intero e value è una stringa alfanumerica di lunghezza arbitraria. Inizialmente le singole coppie sono memorizzate in insiemi disgiunti i quali sono memorizzati in una hash table. Si progetti ed implementi una struttura dati, composta da un hash table in cui ogni cella punta ad un insieme disgiunto, che preveda i seguenti metodi: MAKE\_SET(), FIND\_SET(), UNION(). Dotare il programma di un menu da cui sia possibile richiamare le suddette operazioni.

### 1.1) Descrizione del problema

Mi viene chiesto di creare un hashtable di cui ogni cella punta all'insieme disgiunto e implementare le funzione make-set(), find-set(), union().

Una tabella di hash è un struttura dati molto efficiente per implementare i dizionari. E' molto importante perché l'inserimento di una chiave all'interno del vettore segui dei criteri ben precisi e da qui nasce la funzione hashing che si voglia usare.

Come funzione hash ho usato il metodo della divisione, consiste in questo: Quando sto inserendo la chiave ad esempio 50 farò chiave modulo con il size del mio hashtable e mi ritornerà il resto della divisione e quello sarà l'indice dove inserirò la chiave.

Questo è molto importante per la ricerca perché ricercare un chiave significa calcolarne il modulo quindi so esattamente dove si trova e quindi mi prendere l'accesso all'elemento con complessità  $O(1)$ .

Mi viene chiesto di implementare le varie funzioni:

MAKE-SET()=Creo una lista con un solo elemento

FIND-SET()=Dato un numero cerco il suo rappresentate

UNION()=Dati due chiavi unisco gli insiemi

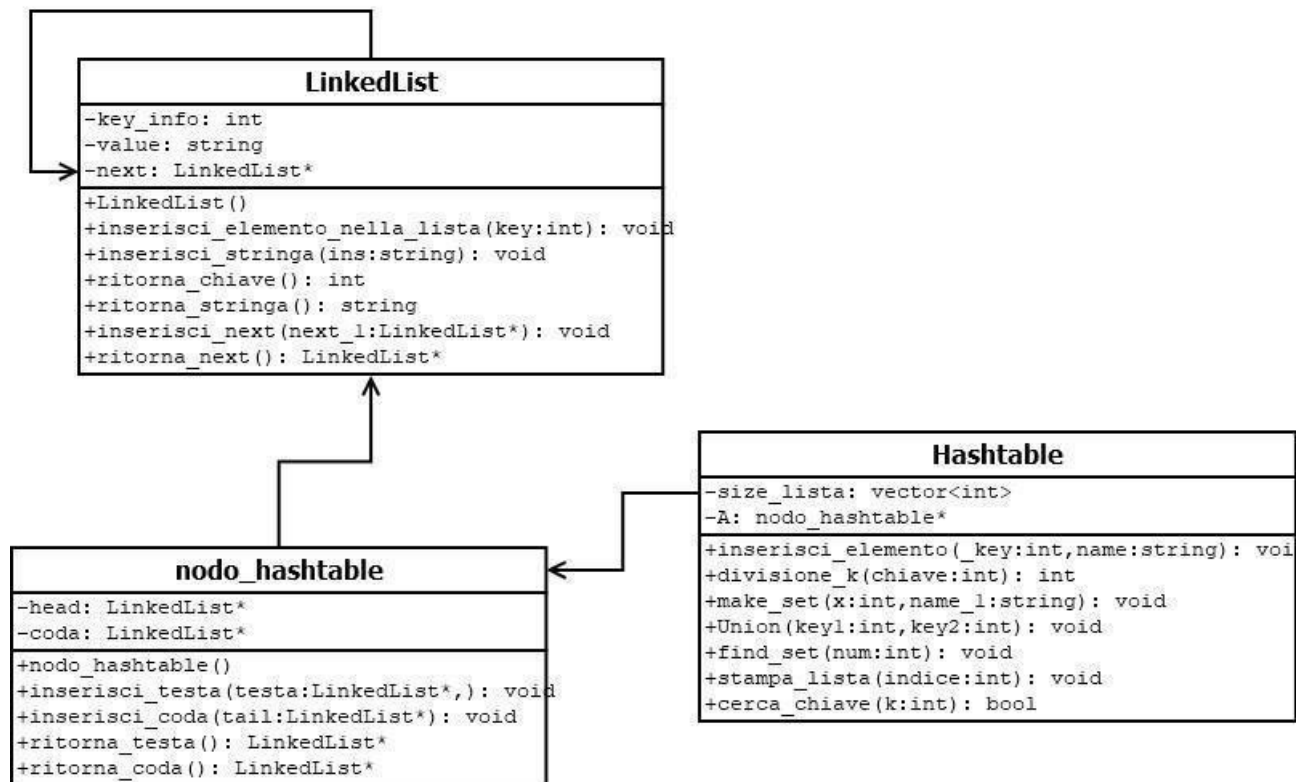
### 1.2) Descrizione strutture dati utilizzate

Come strutture dati ho utilizzato per la costruzione dell'hashtable, una lista che mi rappresenta l'insieme disgiunto e un vettore dove inserirò i vari insiemi disgiunti nella posizione corretta del vettore dato dal criterio della funzione hash.

### 1.3) Formato dati in input/output

Leggo un file input in cui ogni rigo è rappresentato da un intero e una stringa.

#### Diagramm class



### 1.4) Descrizione algoritmo

L'algoritmo funziona in questo modo:

Leggo da un file in input le chiavi e le stringhe, per ogni chiave userò la funzione hash (metodo della divisione) controllerò se la cella corrispondente al modulo della funzione hash è già occupata o meno indipendentemente se è occupata o meno mi creò un oggetto LinkedList che contiene la chiave la stringa e un puntatore.

Se la cella non è occupata allora mi creerò l'oggetto hashtable che praticamente è come se fosse un nodo sentinella che punta alla testa e alla coda della lista e inserirò l'indirizzo dell'oggetto all'interno della cella.

Se la cella è già occupata allora mi dichiaro una variabile temporanea in cui mi salvo il valore della testa della lista corrente dal nodo sentinella e scorrerò la lista fino a trovare il null e inserirò l'elemento (quindi è un inserimento in coda) e ogni volta aggiornerò la nuova coda, per ogni lista che creo avrò un vector che mi conterrà gli elementi presenti nella lista, questo mi servirà per la union.

Il primo elemento inserito nella cella sarà il rappresentante dell'insieme.

La union funziona in questo modo, praticamente avremo un menu dove potremmo inserire le due chiavi che vogliamo inserire, effettuerò un controllo per vedere se le chiavi che sto cercando esistono e se le chiavi che sto cercando di inserire non siano uguali,

quindi con l'informazione che ho nel vector che tiene conto del size delle liste, unirò la lista con il size minore a quella con il size maggiore in modo da diminuire le operazioni di inserimento.

Quindi semplicemente mi scorrerò la lista con size minore e la inserirò nella coda della lista con il size maggiore.

Con la union nasce un problema una volta che ho inserito la lista 2 nella lista 1, se sto cercando ad esempio una chiave e quest'ultima si troverebbe nella lista 2 non la troverò perché ho inserito la lista 2 nella lista 1 e ciò comporterebbe un errore dunque farò puntare la testa della lista 2 alla testa della lista 1.

La find-set() vuole che dato una chiave restituisco il suo rappresentante.

Quindi inserisco la chiave, con il metodo della divisione capirò in quale insieme disgiunto si trova, scorrerò la lista se trovo la chiave restituirò la testa della lista ossia dove si trova il rappresentante.

La make-set() vuole creare una lista con un solo elemento.

Quindi inserisco la chiave e la stringa, controllerò con la funzione hash della chiave se nell'indice restituito se il vettore con quell'indice è già occupato da un altro insieme, se quella cella è vuota creerò il nuovo insieme disgiunto, se invece è occupata la make-set fallisce ma effettuerà un controllo e ci consiglierà la prima cella libera dove poter effettuare una make-set

## 1.5) Complessità

La complessità è data dalle operazioni che effettuiamo nell'hashtable.

Per popolare l'hashtable avremo una complessità data dalla funzione hash, nel caso migliore l'inserimento è  $O(1)$  se la cella è vuota, nel caso peggiore è  $O(n)$  in quanto la cella è già occupata e dobbiamo scorrere la lista e inserire il nuovo elemento in coda.

L'operazione find-set avrà come complessità  $O(1)$  nel momento in cui il find-set della chiave è la chiave stessa, ossia la chiave che sto cercando è anche il rappresentante dell'insieme, mentre negli altri casi avrà complessità  $O(n)$  perché prima devo cercare se una determinata chiave esiste nell'insieme e dopo restituisco il rappresentante che è sempre l'operazione meno costosa in quanto è la prima chiave della lista.

La make-set avrà complessità  $O(1)$  poiché vuole creare un insieme con un solo

Rappresentante quindi si tratterà di fare un inserimento con complessità  $O(1)$  tuttavia se Avrò una complessità  $O(n)$  in quanto mi scorro l'hashtable fino a trovare una cella dove posso effettuare la make-set

Infine abbiamo la union, per migliorare la complessità abbiamo un vector in cui abbiamo i size delle varie liste, quando voglio unire due liste verifico quale sia la lista più piccola e quest'ultima verrà inserita nella coda della lista più grande, l'inserimento in questo caso è  $O(n)$ .

Poiché inserirò  $N$  elementi e l'inserimento nel caso peggiore ha complessità  $O(n)$ , la complessità

finale è  $O(n^2)$ .

## 1.6) Test/risultati

Test con il file input.txt

25,ciao

26,lala

35,qua

90,cias

20,giorgi

50,francesco

2,salvatore

10,ferdinando

21,giovanni

47,paolo

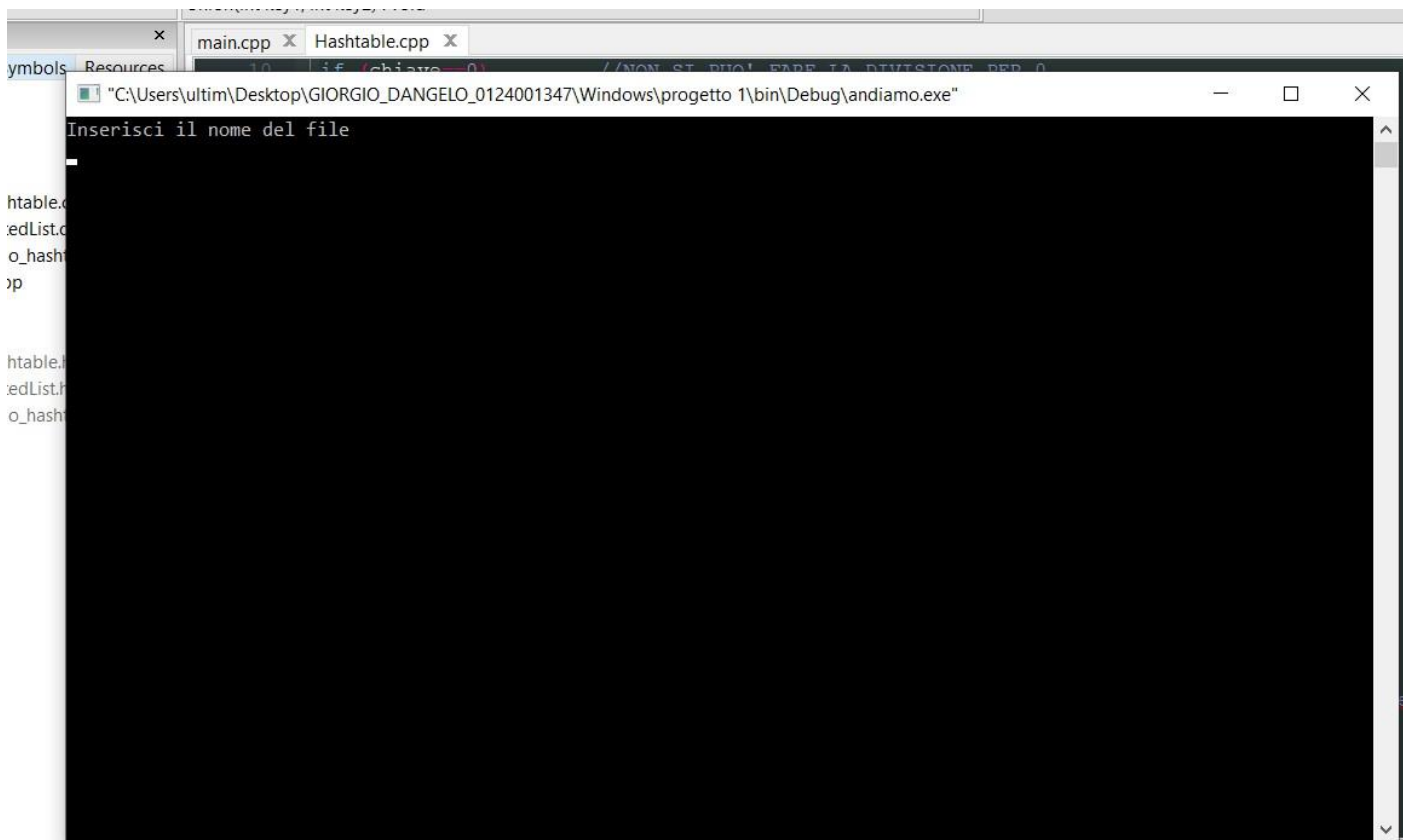
94,luca

703,francesco

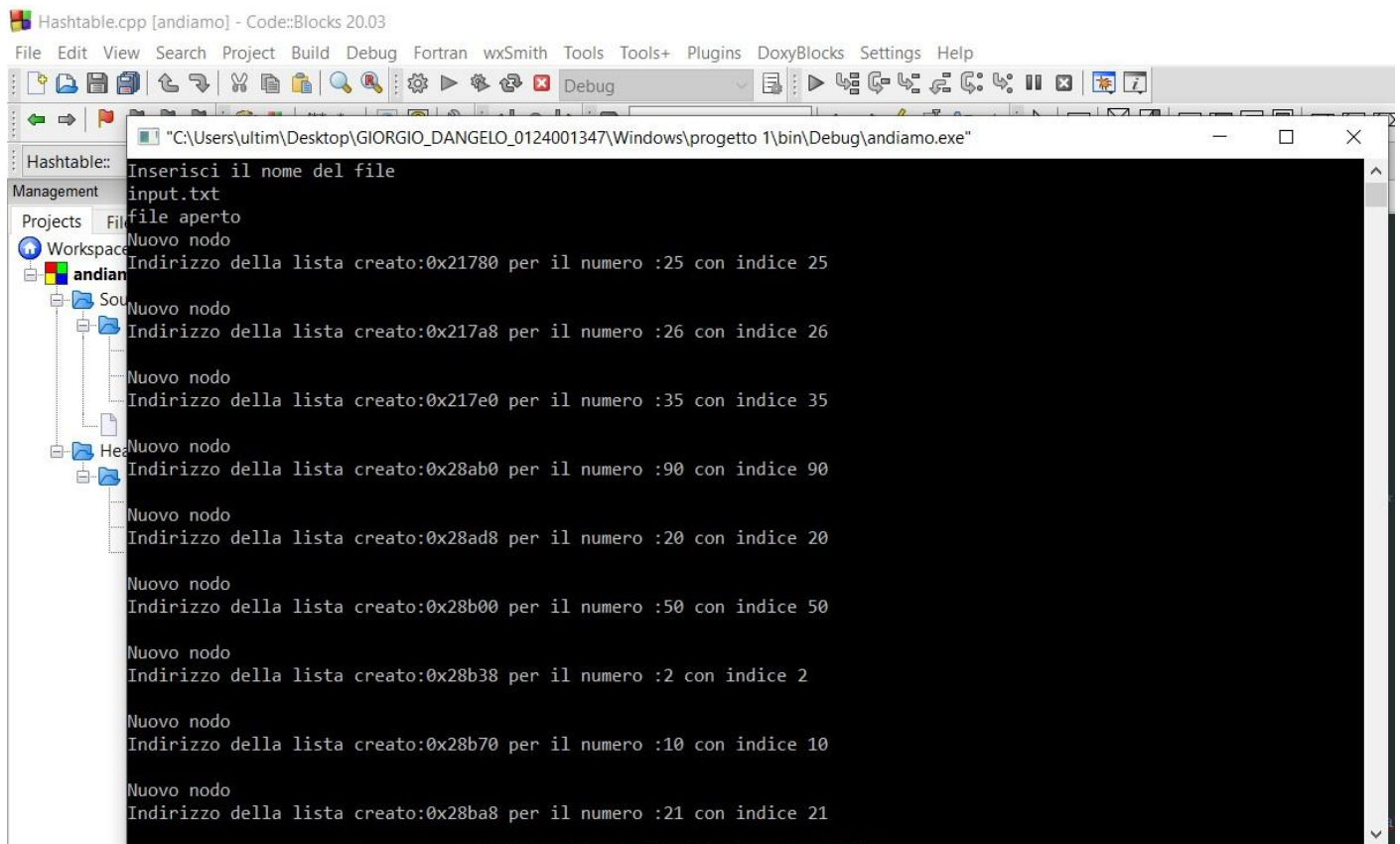
1,paola

702,daniela

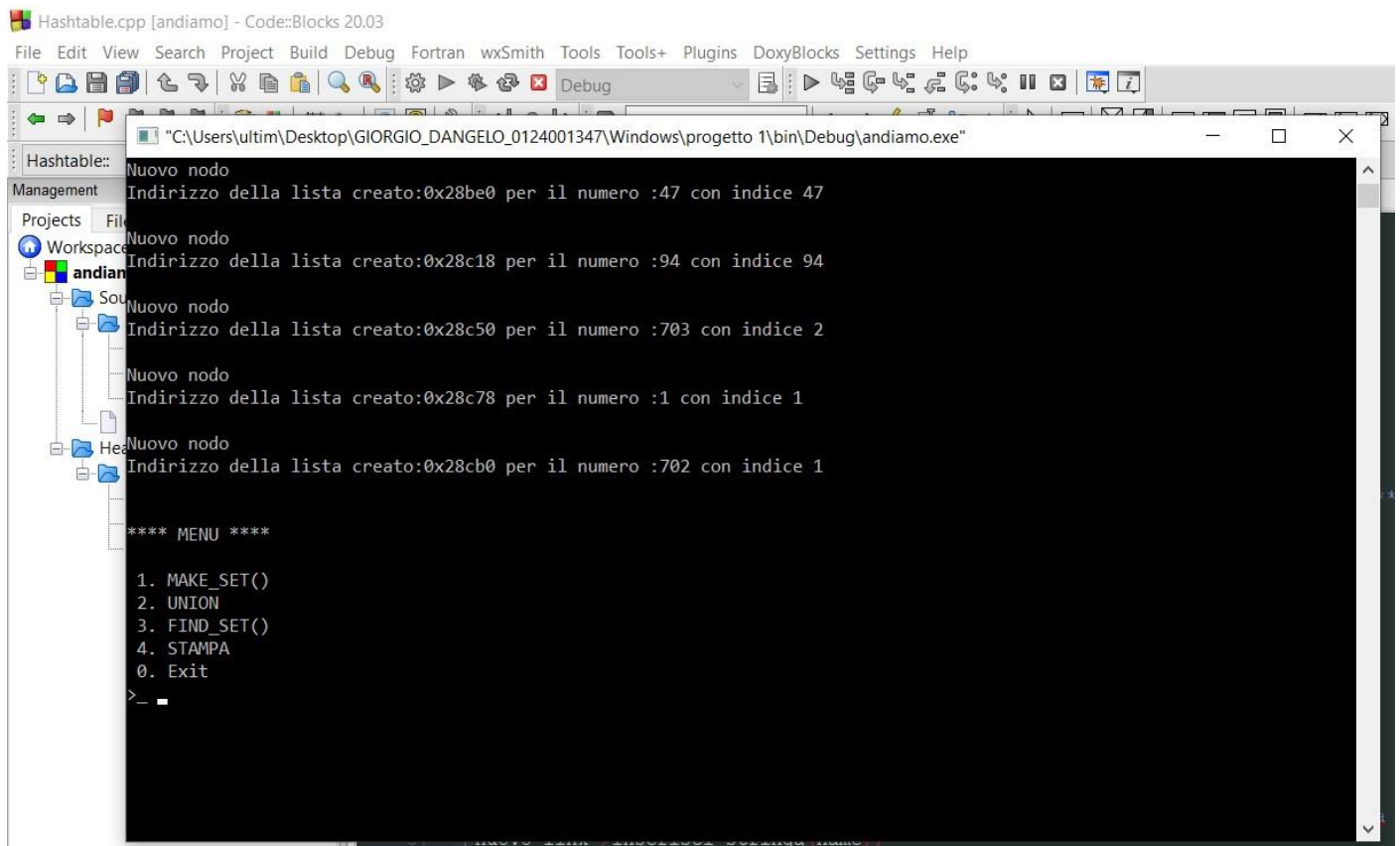
Nel primo passo inseriremo il nome del file in questo caso input.txt



Andremo a vedere gli inserimenti all'interno dell'hashtable



```
Hashtable.cpp [andiamo] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"
Inserisci il nome del file
input.txt
file aperto
Nuovo nodo
Indirizzo della lista creato:0x21780 per il numero :25 con indice 25
Nuovo nodo
Indirizzo della lista creato:0x217a8 per il numero :26 con indice 26
Nuovo nodo
Indirizzo della lista creato:0x217e0 per il numero :35 con indice 35
Nuovo nodo
Indirizzo della lista creato:0x28ab0 per il numero :90 con indice 90
Nuovo nodo
Indirizzo della lista creato:0x28ad8 per il numero :20 con indice 20
Nuovo nodo
Indirizzo della lista creato:0x28b00 per il numero :50 con indice 50
Nuovo nodo
Indirizzo della lista creato:0x28b38 per il numero :2 con indice 2
Nuovo nodo
Indirizzo della lista creato:0x28b70 per il numero :10 con indice 10
Nuovo nodo
Indirizzo della lista creato:0x28ba8 per il numero :21 con indice 21
```



```
Hashtable.cpp [andiamo] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"
Nuovo nodo
Indirizzo della lista creato:0x28be0 per il numero :47 con indice 47
Nuovo nodo
Indirizzo della lista creato:0x28c18 per il numero :94 con indice 94
Nuovo nodo
Indirizzo della lista creato:0x28c50 per il numero :703 con indice 2
Nuovo nodo
Indirizzo della lista creato:0x28c78 per il numero :1 con indice 1
Nuovo nodo
Indirizzo della lista creato:0x28cb0 per il numero :702 con indice 1
**** MENU ****
1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
> _
```

## MAKE-SET

Adesso scegliamo quale opzione eseguire, sceglierò la make-set e voglio inserire la chiave 5 e stringa ciao

```
Hashtable.cpp [andiamo] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"

Indirizzo della lista creato:0x1038b70 per il numero :10 con indice 10
Nuovo nodo
Indirizzo della lista creato:0x1038ba8 per il numero :21 con indice 21
Nuovo nodo
Indirizzo della lista creato:0x1038be0 per il numero :47 con indice 47
Nuovo nodo
Indirizzo della lista creato:0x1038c18 per il numero :94 con indice 94
Nuovo nodo
Indirizzo della lista creato:0x1038c50 per il numero :703 con indice 2
Nuovo nodo
Indirizzo della lista creato:0x1038c78 per il numero :1 con indice 1
Nuovo nodo
Indirizzo della lista creato:0x1038cb0 per il numero :702 con indice 1

**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_ 1
MAKE_SET() INSERISCI LA CHIAVE E LA STRINGA DA INSERIRE :5 ciao
```

Ecco l'inserimento

```
Hashtable.cpp [andiamo] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"

Nuovo nodo
Indirizzo della lista creato:0x1038c50 per il numero :703 con indice 2
Nuovo nodo
Indirizzo della lista creato:0x1038c78 per il numero :1 con indice 1
Nuovo nodo
Indirizzo della lista creato:0x1038cb0 per il numero :702 con indice 1

**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_ 1
MAKE_SET() INSERISCI LA CHIAVE E LA STRINGA DA INSERIRE :5 ciao
Insieme disgiunto creato nell'indice di posizione 5 del numero :5

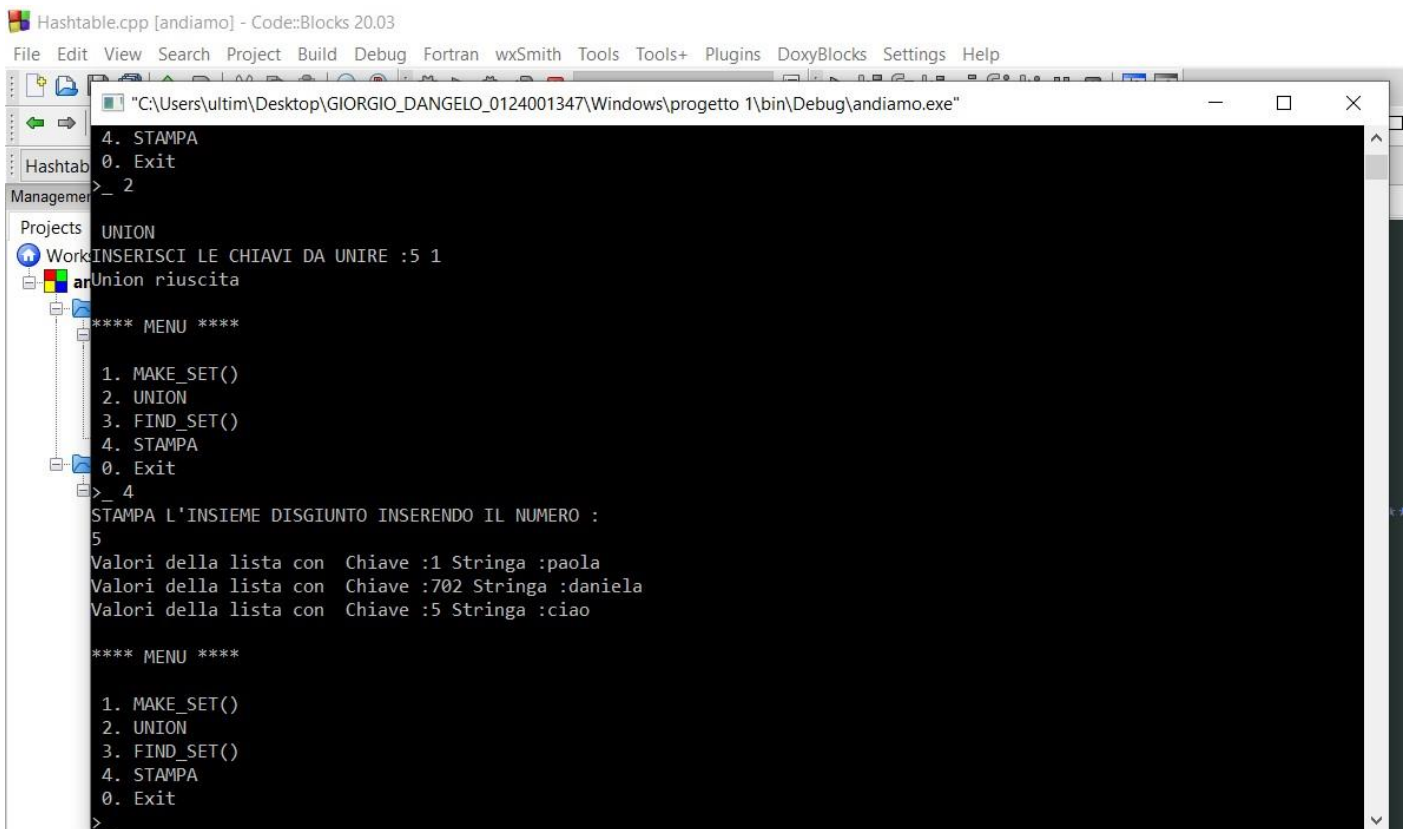
**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>
```



# UNION

Sceglierò di fare la union inserendo due chiavi, 5 e 1



```
Hashtable.cpp [andiamo] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"

4. STAMPA
0. Exit
>_ 2
Union riuscita

**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_ 4
STAMPA L'INSIEME DISGIUNTO INSERENDO IL NUMERO :
5
Valori della lista con Chiave :1 Stringa :paola
Valori della lista con Chiave :702 Stringa :daniela
Valori della lista con Chiave :5 Stringa :ciao

**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_
```

La union è andata a buon fine e sono state unite le due liste poiché il size della lista di 1 è maggiore rispetto a quello della lista di 5 in quanto contiene un solo elemento ,la lista 5 è stata unita in coda alla lista 1

Find-set: voglio il rappresentante di 5



Hashtable.cpp [andiamo] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

```
"C:\Users\ultim\Desktop\GIORGIO_DANGELO_0124001347\Windows\progetto 1\bin\Debug\andiamo.exe"

2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_ 4
STAMPA L'INSIEME DISGIUNTO INSERENDO IL NUMERO :
5
Valori della lista con Chiave :1 Stringa :paola
Valori della lista con Chiave :702 Stringa :daniela
Valori della lista con Chiave :5 Stringa :ciao

**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>_ 3
FIND_SET() INSERISCI IL NODO DI CUI VUOI TROVARE IL RAPPRESENTANTE :
5
Il rappresentante di 5 e' :1
**** MENU ****

1. MAKE_SET()
2. UNION
3. FIND_SET()
4. STAMPA
0. Exit
>
```