



POLITECNICO  
DI MILANO

# INFORMATICA

Struttura di un  
programma per tabelle  
rappresentate con array  
di record

## Tabella ordinata

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inserimento
- Eliminazione
- Stampa

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inserimento
- Eliminazione
- Stampa

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75



**Campo chiave**

Definisce univocamente

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
84	Gialli	Carla	f	5	11	75
48	Neri	Anna	f	4	7	75
7	Rossi	Marco	m	5	10	75
63	Verdi	Remo	m	5	8	76

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
84	Gialli	Carla	f	5	11	75
48	Neri	Anna	f	4	7	75
7	Rossi	Marco	m	5	10	75
93	Rossi	Mario	m	12	3	78
63	Verdi	Remo	m	5	8	76

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inizializzazione della tabella
- Inserimento record
- Eliminazione record

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inizializzazione della tabella
- Inserimento record
- Eliminazione record
- Fine





## Memorizzare su file


Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75



## Recuperare da file

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75





Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75
15	Bianchi	Paolo	m	12	6	78

- Inizializzazione della tabella
- **Inserimento record**
- Eliminazione record
- Fine

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
15	Bianchi	Paolo	m	12	6	78
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inizializzazione della tabella
- **Inserimento record**
- Eliminazione record
- Fine

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	<del>Anna</del>	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

- Inizializzazione della tabella
- Inserimento record
- **Eliminazione record**
- Fine

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

Diagram illustrating a table structure with records. The top table shows a record for Marco Rossi (Matr. 7, Data nascita 5/10/75). The bottom table shows records for Remo Verdi (Matr. 63, Data nascita 5/8/76) and Carla Gialli (Matr. 84, Data nascita 5/11/75). Red arrows indicate the mapping of the top record's fields to the corresponding fields in the bottom table.

- Inizializzazione della tabella
- Inserimento record
- **Eliminazione record**
- Fine

## Problemi di efficienza


Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

*funzione1()*

- Inizializzazione della tabella
- **Inserimento record**
- Eliminazione record
- Fine



Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75

63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

*funzione2()*

- Inizializzazione della tabella
- Inserimento record
- **Eliminazione record**
- Fine

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

*funzione2()*

- Inizializzazione della tabella
- Inserimento record
- **Eliminazione record**
- Fine

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	63	Verdi	Remo	m	5	8	76
2	84	Gialli	Carla	f	5	11	75
3							

*num*

- Inizializzazione della tabella
- Inserimento record
- **Eliminazione record**
- Fine

```
main()  
{ inizializzaTabella(...);
```



num	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
4	7	Rossi	Marco	m	5	10	75
	48	Neri	Anna	f	4	7	75
	63	Verdi	Remo	m	5	8	76
	84	Gialli	Carla	f	5	11	75

```
main()  
{ inizializzaTabella(...);  
  while(ci sono aggiornamenti)  
  { se l'aggiornamento è un inserimento  
    inserisciSeNonEsiste(...);
```

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
<b>48</b>	<b>Neri</b>	<b>Anna</b>	<b>f</b>	<b>4</b>	<b>7</b>	<b>75</b>
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75
<b>48</b>	<b>Neri</b>	<b>Anna</b>	<b>f</b>	<b>4</b>	<b>7</b>	<b>75</b>

```
main()  
{ inizializzaTabella(...);  
  while(ci sono aggiornamenti)  
  { se l'aggiornamento è un inserimento  
    inserisciSeNonEsiste(...);  
    se l'aggiornamento è un'eliminazione
```

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

```
main()  
{ inizializzaTabella(...);  
  while(ci sono aggiornamenti)  
  { se l'aggiornamento è un inserimento  
    inserisciSeNonEsiste(...);  
    se l'aggiornamento è un'eliminazione  
    eliminaSeEsiste(...);  
  }  
}
```

```
main()  
{ inizializzaTabella(...) ;  
  while(ci sono aggiornamenti)  
    { se l'aggiornamento è un inserimento  
      inserisciSeNonEsiste(...) ;  
      se l'aggiornamento è un'eliminazione  
      eliminaSeEsiste(...) ;  
    }  
}
```



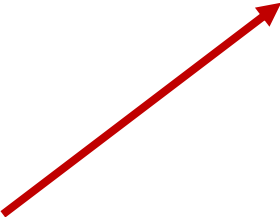
```
main()
{
    inizializzaTabella(...);
    stampaTabella(...);
    while(ci sono aggiornamenti)
    {
        se l'aggiornamento è un inserimento
        {
            inserisciSeNonEsiste(...);
            stampaTabella(...);
        }

        se l'aggiornamento è un'eliminazione
        {
            eliminaSeEsiste(...);
            stampaTabella(...);
        }
    }
}
```

## Intestazioni

```
? inizializzaTabella (?);  
? inserisciSeNonEsiste (?);  
? eliminaSeEsiste (?);  
? stampaTabella (?);
```

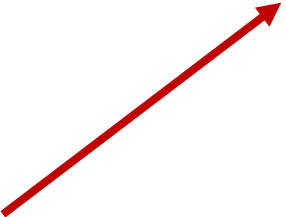
```
int main()  
{ ...  
}
```



## Ingresso

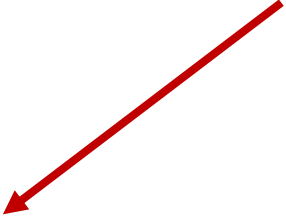
```
... inizializzaTabella(tipo, tipo &);  
... inserisciSeNonEsiste(tipo, tipo &);  
... eliminaSeEsiste(tipo, tipo &);  
... stampaTabella(tipo, tipo &);
```

```
int main()  
{ ...  
}
```



## Uscita

```
...  inizializzaTabella(tipo, tipo &);  
...  inserisciSeNonEsiste(tipo, tipo &);  
...  eliminaSeEsiste(tipo, tipo &);  
...  stampaTabella(tipo, tipo &);
```



```
int main()  
{ ...  
}
```

```
... inizializzaTabella(tipo, tipo &);
```



**struct InsStudenti**

num
4

**0**  
**1**  
**2**  
**3**

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

```
void  inizializzaTabella (InsStudenti &);
```

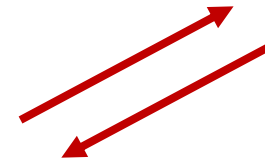
```
tipo inizializzaTabella() ;
```

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);
```

num
4

0  
1  
2  
3

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75





## Passaggio per indirizzo

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);
```

Per valore

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);
```

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);
```

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);
```

num	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
4	7	Rossi	Marco	m	5	10	75
	48	Neri	Anna	f	4	7	75
	63	Verdi	Remo	m	5	8	76
	84	Gialli	Carla	f	5	11	75

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);
```

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```



cout

num
4

0

1

2

3

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```



cout

num
4

0

1

2

3

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75



## Prototipi

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```

## Efficienza

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```

num
4

0

1

2

3

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A
7	Rossi	Marco	m	5	10	75
48	Neri	Anna	f	4	7	75
63	Verdi	Remo	m	5	8	76
84	Gialli	Carla	f	5	11	75

# Efficienza

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);
```

## Programma

*Ambiente globale*  
*(+ prototipi)*

```
main()
```

```
{...}
```

```
InsStudenti inizializzaTabella()
```

```
{...}
```

```
void inserisciSeNonEsiste(...)
```

```
{...}
```

```
void eliminaSeEsiste(...)
```

```
{...}
```

```
void stampaTabella(...)
```

```
{...}
```

*Ambiente globale*  
*(+ prototipi)*

**main()**

{...}

InsStudenti inizializzaTabella()

{...}

void inserisciSeNonEsiste(...)

{...}

void eliminaSeEsiste(...)

{...}

void stampaTabella(...)

{...}

*Ambiente globale*



*Tipi*

*(+ propotipi)*

```
main()
```

```
{...}
```

```
InsStudenti inizializzaTabella()
```

```
{...}
```

```
void inserisciSeNonEsiste(...)
```

```
{...}
```

```
void eliminaSeEsiste(...)
```

```
{...}
```

```
void stampaTabella(...)
```

```
{...}
```

```
#include <iostream>
using namespace std;
#include <iomanip.h>

const int FALSO = 0;
const int VERO = 1;
struct Data
{
    int giorno,          // 1-31
    mese,                // 1-12
    anno;                // 00-99
};
const int MAXIDENT = 10;
struct Studente
{
    int matricola;
    char cognome[MAXIDENT],
        nome[MAXIDENT],
        sesso;           // M,F
    Data dataNascita;
};
const int DIM = 8;
struct InsStudenti
{
    int num;              // numero di righe significative
    Studente elenco[DIM];
};
```

```
#include <iostream>
using namespace std;
#include <iomanip.h>

const int FALSO = 0;
const int VERO = 1;
struct Data
{
    int giorno,          // 1-31
    mese,                // 1-12
    anno;                // 00-99
};
const int MAXIDENT = 10;
struct Studente
{
    int matricola;
    char cognome[MAXIDENT],
        nome[MAXIDENT],
        sesso;           // M,F
    Data dataNascita;
};
const int DIM = 8;
struct InsStudenti
{
    int num;              // numero di righe significative
    Studente elenco[DIM];
};
```



```
int main()
```

```
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;           // operaz. da effettuare sulla tabella
  Studente nuovoStudente;    // dati dello studente da inserire
  int matricola;             // matric. dello studente da eliminare
  InsStudenti classe;
  //inizializzazione della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  //acquisizione e gestione delle operazioni sulla tabella
  //intestazione in uscita
  cout << setw(50) << "acquisizione operazioni" << endl << endl;
  //acquisizione di operazione per prima rilevazione
  cout << "operazione? (i per inserimento,"
        << " e per eliminazione, f per fine): "
  cin >> operazione;
  while (operazione != FINE)
  { if (operazione == INSERISCI)
    { //lettura dei dati del nuovo studente
      cout << " dati del nuovo studente" << endl
      cout << " matricola: "; cin >> nuovoStudente.matricola;
      cout << " cognome: "; cin >> nuovoStudente.cognome;
      cout << " nome: "; cin >> nuovoStudente.nome;
      cout << " sesso(M/F): "; cin >> nuovoStudente.sesso;
```

## Carattere

```
const char INSERISCI = 'i';
const char ELIMINA = 'e';
const char FINE = 'f';
char operazione;           // operaz. da effettuare sulla tabella
Studente nuovoStudente;   // dati dello studente da inserire
int matricola;             // matric. dello studente da eliminare
InsStudenti classe;
//inizializzazione della tabella
classe = inizializzaTabella();
//acquisizione delle operazioni sulla tabella
//input con uscita
cout << "0) << \"acquisizione operazioni\" << endl << endl;
//acquisizione operazione per prima rilevazione
cout << "operazione? (i per inserimento, "
    << " e per eliminazione, f per fine): "
    << endl;
cin >> operazione;
while (operazione != FINE)
{ if (operazione == INSERISCI)
    { //lettura dei dati del nuovo studente
      cout << " dati del nuovo studente" << endl
        << " matricola: "; cin >> nuovoStudente.matricola;
        cout << " cognome: "; cin >> nuovoStudente.cognome;
        cout << " nome: "; cin >> nuovoStudente.nome;
        cout << " sesso(M/F): "; cin >> nuovoStudente.sesso;
```



- Inserimento
- Eliminazione
- Fine

```
Int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;           // operaz. da effettuare sulla tabella
  Studente nuovoStudente;    // dati dello studente da inserire
  int matricola;             // matric. dello studente da eliminare
  InsStudenti classe;
  //inizializzazione della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  //acquisizione delle operazioni sulla tabella
  //input con uscita
  cout << "0) << \"acquisizione operazioni\" << endl << endl;
  //acquisizione di operazione per prima rilevazione
  cout << "operazione? (i per inserimento, "
        << " e per eliminazione, f per fine): "
        << endl;
  cin >> operazione;
  while (operazione != FINE)
  { if (operazione == INSERISCI)
    { //lettura dei dati del nuovo studente
      cout << " dati del nuovo studente" << endl
      cout << " matricola: "; cin >> nuovoStudente.matricola;
      cout << " cognome: "; cin >> nuovoStudente.cognome;
      cout << " nome: "; cin >> nuovoStudente.nome;
      cout << " sesso(M/F): "; cin >> nuovoStudente.sesso;
```

**i** • Inserimento  
**e** • Eliminazione  
**f** • Fine



```
int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;           // operaz. da effettuare sulla tabella
  Studente nuovoStudente;    // dati dello studente da inserire
  int matricola;             // matric. dello studente da eliminare
  InsStudenti classe;
  //inizializzazione della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  //acquisizione e gestione delle operazioni sulla tabella
  //intestazione in uscita
  cout << setw(50) << "acquisizione operazioni" << endl << endl;
  //acquisizione di operazione per prima rilevazione
  cout << "operazione? (i per inserimento,"
        << " e per eliminazione, f per fine): "
  cin >> operazione;
  while (operazione != FINE)
  { if (operazione == INSERISCI)
    { //lettura dei dati del nuovo studente
      cout << " dati del nuovo studente" << endl
      cout << " matricola: "; cin >> nuovoStudente.matricola;
      cout << " cognome: "; cin >> nuovoStudente.cognome;
      cout << " nome: "; cin >> nuovoStudente.nome;
      cout << " sesso(M/F): "; cin >> nuovoStudente.sesso;
```

```
    stampaTabella(classe);  
//acquisizione e gestione delle operazioni sulla tabella  
//intestazione in uscita  
    cout << setw(50) << "acquisizione operazioni" << endl << endl;  
//acquisizione di operazione per prima rilevazione  
    cout << "operazione? (i per inserimento,"  
        << " e per eliminazione, f per fine): "  
    cin >> operazione;  
    while (operazione != FINE)  
    { if (operazione == INSERISCI)  
        { //lettura dei dati del nuovo studente  
            cout << " dati del nuovo studente" << endl  
            cout << " matricola: "; cin >> nuovoStudente.matricola;  
            cout << " cognome: "; cin >> nuovoStudente.cognome;  
            cout << " nome: "; cin >> nuovoStudente.nome;  
            cout << " sesso(M/F): "; cin >> nuovoStudente.sesso;  
            cout << " giorno di nascita (1-31): ";  
            cin >> nuovoStudente.dataNascita.giorno;  
            cout << " mese di nascita (1-12): ";  
            cin >> nuovoStudente.dataNascita.mese;  
            cout << "anno di nascita (0-99): ";  
            cin >> nuovoStudente.dataNascita.anno;  
            inserisciSeNonEsiste(nuovoStudente, classe);  
            stampaTabella(classe);  
        }  
        else if (operazione == ELIMINA)
```

```
struct Studente
{ int matricola;
  char cognome[MAXIDENT],
    nome[MAXIDENT],
    sesso;          // M,F
  Data dataNascita;
};
```

```
{ //lettura dei dati del nuovo studente
  cout << " dati del nuovo studente" << endl
  cout << " matricola: "; cin >> nuovoStudente.matricola;
  cout << " cognome: "; cin >> nuovoStudente.cognome;
  cout << " nome: "; cin >> nuovoStudente.nome;
  cout << " sesso (M/F): "; cin >> nuovoStudente.sesso;
  cout << " giorno di nascita (1-31): ";
  cin >> nuovoStudente.dataNascita.giorno;
  cout << " mese di nascita (1-12): ";
  cin >> nuovoStudente.dataNascita.mese;
  cout << "anno di nascita (0-99): ";
  cin >> nuovoStudente.dataNascita.anno;
  inserisciSeNonEsiste(nuovoStudente, classe);
  stampaTabella(classe);
}
else if (operazione == ELIMINA)
```

```
struct Data
{ int giorno,      // 1-31
  mese,           // 1-12
  anno;           // 00-99
};
```

```
{ //lettura dei dati del nuovo studente
  cout << " dati del nuovo studente" << endl
  cout << " matricola: "; cin >> nuovoStudiante.matricola;
  cout << " cognome: "; cin >> nuovoStudiante.cognome;
  cout << " nome: "; cin >> nuovoStudiante.nome;
  cout << " sesso (M/F): "; cin >> nuovoStudiante.sesso;
  cout << " giorno di nascita (1-31): ";
  cin >> nuovoStudiante.dataNascita.giorno;
  cout << " mese di nascita (1-12): ";
  cin >> nuovoStudiante.dataNascita.mese;
  cout << "anno di nascita (0-99): ";
  cin >> nuovoStudiante.dataNascita.anno;
  inserisciSeNonEsiste(nuovoStudiante, classe);
  stampaTabella(classe);
}
else if (operazione == ELIMINA)
```

```
    stampaTabella(classe);  
//acquisizione e gestione delle operazioni sulla tabella  
//intestazione in uscita  
    cout << setw(50) << "acquisizione operazioni" << endl << endl;  
//acquisizione di operazione per prima rilevazione  
    cout << "operazione? (i per inserimento,"  
        << " e per eliminazione, f per fine): "  
    cin >> operazione;  
while (operazione != FINE)  
{ if (operazione == INSERISCI)  
    { //lettura dei dati del nuovo studente  
        cout << " dati del nuovo studente" << endl  
        cout << " matricola: "; cin >> nuovoStudiante.matricola;  
        cout << " cognome: "; cin >> nuovoStudiante.cognome;  
        cout << " nome: "; cin >> nuovoStudiante.nome;  
        cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;  
        cout << " giorno di nascita (1-31): ";  
        cin >> nuovoStudiante.dataNascita.giorno;  
        cout << " mese di nascita (1-12): ";  
        cin >> nuovoStudiante.dataNascita.mese;  
        cout << "anno di nascita (0-99): ";  
        cin >> nuovoStudiante.dataNascita.anno;  
        inserisciSeNonEsiste(nuovoStudiante, classe);  
        stampaTabella(classe);  
    }  
    else if (operazione == ELIMINA)
```



```
    stampaTabella(classe);  
//acquisizione e gestione delle operazioni sulla tabella  
//intestazione in uscita  
    cout << setw(50) << "acquisizione operazioni" << endl << endl;  
//acquisizione di operazione per prima rilevazione  
    cout << "operazione? (i per inserimento,"  
        << " e per eliminazione, f per fine): "  
    cin >> operazione;  
while (operazione != FINE)  
    { if (operazione == INSERISCI)  
        { //lettura dei dati del nuovo studente  
            cout << " dati del nuovo studente" << endl  
            cout << " matricola: "; cin >> nuovoStudiante.matricola;  
            cout << " cognome: "; cin >> nuovoStudiante.cognome;  
            cout << " nome: "; cin >> nuovoStudiante.nome;  
            cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;  
            cout << " giorno di nascita (1-31): ";  
            cin >> nuovoStudiante.dataNascita.giorno;  
            cout << " mese di nascita (1-12): ";  
            cin >> nuovoStudiante.dataNascita.mese;  
            cout << "anno di nascita (0-99): ";  
            cin >> nuovoStudiante.dataNascita.anno;  
            inserisciSeNonEsiste(nuovoStudiante, classe);  
            stampaTabella(classe);  
        }  
        else if (operazione == ELIMINA)
```

```
struct Data
{ int giorno,      // 1-31
  mese,           // 1-12
  anno;           // 00-99
};
```

```
{ //lettura dei dati del nuovo studente
  cout << " dati del nuovo studente" << endl
  cout << " matricola: "; cin >> nuovoStudiante.matricola;
  cout << " cognome: "; cin >> nuovoStudiante.cognome;
  cout << " nome: "; cin >> nuovoStudiante.nome;
  cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;
  cout << " giorno di nascita (1-31): ";
  cin >> nuovoStudiante.dataNascita.giorno;
  cout << " mese di nascita (1-12): ";
  cin >> nuovoStudiante.dataNascita.mese;
  cout << "anno di nascita (0-99): ";
  cin >> nuovoStudiante.dataNascita.anno;
  inserisciSeNonEsiste(nuovoStudiante, classe);
  stampaTabella(classe);
}
else if (operazione == ELIMINA)
```

```
    stampaTabella(classe);  
//acquisizione e gestione delle operazioni sulla tabella  
//intestazione in uscita  
    cout << setw(50) << "acquisizione operazioni" << endl << endl;  
//acquisizione di operazione per prima rilevazione  
    cout << "operazione? (i per inserimento,"  
        << " e per eliminazione, f per fine): "  
    cin >> operazione;  
while (operazione != FINE)  
    { if (operazione == INSERISCI)  
        { //lettura dei dati del nuovo studente  
            cout << " dati del nuovo studente" << endl  
            cout << " matricola: "; cin >> nuovoStudiante.matricola;  
            cout << " cognome: "; cin >> nuovoStudiante.cognome;  
            cout << " nome: "; cin >> nuovoStudiante.nome;  
            cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;  
            cout << " giorno di nascita (1-31): ";  
            cin >> nuovoStudiante.dataNascita.giorno;  
            cout << " mese di nascita (1-12): ";  
            cin >> nuovoStudiante.dataNascita.mese;  
            cout << "anno di nascita (0-99): ";  
            cin >> nuovoStudiante.dataNascita.anno;  
            inserisciSeNonEsiste(nuovoStudiante, classe);  
            stampaTabella(classe);  
        }  
        else if (operazione == ELIMINA)
```

```
    cout << " nome: "; cin >> nuovoStudiante.nome;
    cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;
    cout << " giorno di nascita (1-31): ";
    cin >> nuovoStudiante.dataNascita.giorno;
    cout << " mese di nascita (1-12): ";
    cin >> nuovoStudiante.dataNascita.mese;
    cout << "anno di nascita (0-99): ";
    cin >> nuovoStudiante.dataNascita.anno;
    inserisciSeNonEsiste(nuovoStudiante, classe);
    stampaTabella(classe);
}
else if (operazione == ELIMINA)
{ //lettura matricola studente da eliminare
    cout << " matricola studente da eliminare: ";
    cin >> matricola;
    eliminaSeEsiste(matricola, classe);
    stampaTabella(classe);
}

//acquisizione di prossima operazione
cout << "operazione? ( i per inserimento,"
<< " e per eliminazione,"
<< " f per fine" << " ): ";
cin >> operazione;
};
}
```

```
        cout << " nome: "; cin >> nuovoStudiante.nome;
        cout << " sesso(M/F): "; cin >> nuovoStudiante.sesso;
        cout << " giorno di nascita (1-31): ";
        cin >> nuovoStudiante.dataNascita.giorno;
        cout << " mese di nascita (1-12): ";
        cin >> nuovoStudiante.dataNascita.mese;
        cout << "anno di nascita (0-99): ";
        cin >> nuovoStudiante.dataNascita.anno;
        inserisciSeNonEsiste(nuovoStudiante, classe);
        stampaTabella(classe);
    }
    else if (operazione == ELIMINA)
    { //lettura matricola studente da eliminare
        cout << " matricola studente da eliminare: ";
        cin >> matricola;
        eliminaSeEsiste(matricola, classe);
        stampaTabella(classe);
    }

    //acquisizione di prossima operazione
    cout << "operazione? ( i per inserimento,"
    << " e per eliminazione,"
    << " f per fine" << " ): ";
    cin >> operazione;
};
}
```