



POLITECNICO
DI MILANO

INFORMATICA

Funzioni per
l'aggiornamento di
tabelle rappresentate
con array di record

ambiente globale

InsStudenti **inizializzaTabella()** ;

void **inserisciSeNonEsiste**(**Studente**, **InsStudenti** &);

void **eliminaSeEsiste**(**int**, **InsStudenti** &);

void **stampaTabella**(**InsStudenti** &);

int **main**()

{...}

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
int main()
```

```
{...
```

```
    InsStudenti classe;
```

```
    classe = inizializzaTabella();
```

```
    ...
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...  
    InsStudenti classe;  
    classe = inizializzaTabella();  
    ...
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
struct InsStudenti  
    { int num;  
      Studente elenco[DIM];  
    };
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
struct InsStudenti  
{ int num;  
  Studente elenco[DIM];  
};
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

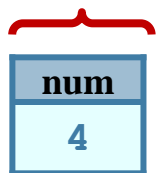
```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
struct InsStudenti  
{ int num;  
  Studente elenco[DIM];  
};
```



num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
int main()
```

```
{...
```

```
    InsStudenti classe;
```

```
    classe = inizializzaTabella();
```

```
    ...
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75


```
InsStudenti inizializzaTabella();  
{  
    ...  
    return valore;  
}
```

num

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}}},  
          {48, "verdi", "anna", 'F', {4, 7, 75}},  
          {63, "neri", "remo", 'M', {5, 8, 76}},  
          {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

num

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

num

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A

Inizializzazione di variabili strutturate

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

num

Matr.	Cognome	Nome	Sesso	Data nascita		
				G	M	A

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● **tab** { ● **num**
 ● **elenco**

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}}},  
          {48, "verdi", "anna", 'F', {4, 7, 75}},  
          {63, "neri", "remo", 'M', {5, 8, 76}},  
          {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}}},  
          {48, "verdi", "anna", 'F', {4, 7, 75}},  
          {63, "neri", "remo", 'M', {5, 8, 76}},  
          {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        {  
            4,  
            {{7, "rossi", "marco", 'M', {5, 10, 76}}},  
            {48, "verdi", "anna", 'F', {4, 7, 75}},  
            {63, "neri", "remo", 'M', {5, 8, 76}},  
            {84, "gialli", "carla", 'F', {5, 11, 75}}  
        }  
    };  
    return tab;  
}
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75


```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}}},  
          {48, "verdi", "anna", 'F', {4, 7, 75}}},  
          {63, "neri", "remo", 'M', {5, 8, 76}}},  
          {84, "gialli", "carla", 'F', {5,11,75}}}  
        };  
    return tab;  
}
```

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

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();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

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();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

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();  
{  
    InsStudenti tab =  
        { 4, {  
            {7, "rossi", "marco", 'M', {5, 10, 76}},  
            {48, "verdi", "anna", 'F', {4, 7, 75}},  
            {63, "neri", "remo", 'M', {5, 8, 76}},  
            {84, "gialli", "carla", 'F', {5, 11, 75}}  
        }  
    };  
    return tab;  
}
```

- `tab` {
 - `num`
 - `elenco`
- `matricola`
 - `cognome`
 - `nome`
 - `Sesso`
 - `dataNascita`

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno


```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5, 10, 76}},  
           {48, "verdi", "anna", 'F', {4, 7, 75}},  
           {63, "neri", "remo", 'M', {5, 8, 76}},  
           {84, "gialli", "carla", 'F', {5, 11, 75}}  
        };  
    return tab;  
}
```

● tab { ● num
 ● elenco { ● matricola
 ● cognome
 ● nome
 ● sesso
 ● dataNascita { ● giorno
 ● mese
 ● anno

```
InsStudenti inizializzaTabella();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

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();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        };  
    return tab;  
}
```

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();  
{  
    InsStudenti tab =  
        { 4,  
          {{7, "rossi", "marco", 'M', {5,10,76}},  
           {48, "verdi", "anna", 'F', {4, 7,75}},  
           {63, "neri", "remo", 'M', {5, 8,76}},  
           {84, "gialli", "carla", 'F', {5,11,75}}  
        }  
    };  
    return tab;  
}
```

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


```
int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;
  Studente nuovoStudente;
  int matricola;
  InsStudenti classe;
  //inizializzazionee della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  ...
```

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

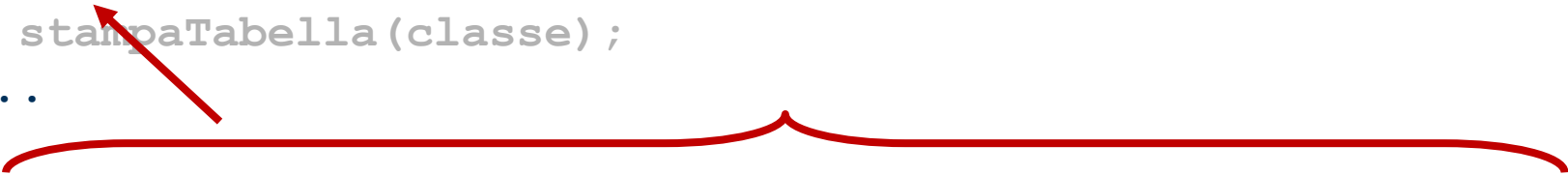
```
int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;
  Studente nuovoStudente;
  int matricola;
  InsStudenti classe;
  //inizializzazionee della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  ...
```

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

```
int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;
  Studente nuovoStudente;
  int matricola;
  InsStudenti classe;
  //inizializzazione della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  ...
```



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

```
int main()
{ const char INSERISCI = 'i';
  const char ELIMINA = 'e';
  const char FINE = 'f';
  char operazione;
  Studente nuovoStudente;
  int matricola;
  InsStudenti classe;
  //inizializzazione della tabella
  classe = inizializzaTabella();
  stampaTabella(classe);
  ...
```

```
void stampaTabella(Tabella &tab)
{ ...
}
```

```
void stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

```
void stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

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 stampaTabella(Tabella &tab)
{
    int i;
    cout << " tabella con il solo campo chiave "
          << endl;
    for(i = 0; i < tab.num; i++)
        cout << setw(5) << tab.elenco[i].matricola;
    cout << endl << endl;
}
```

num
4

0
1
...
i

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

ambiente globale

Ricerca

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

Ricerca

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

Ricerca sequenziale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

Ricerca sequenziale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void ricerca binaria/logaritmica (InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

Ricerca sequenziale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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 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

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

91

↓

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 inserisciSeNonEsiste(Studente, InsStudenti &);
```

91



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 inserisciSeNonEsiste(Studente, InsStudenti &);
```

num
4

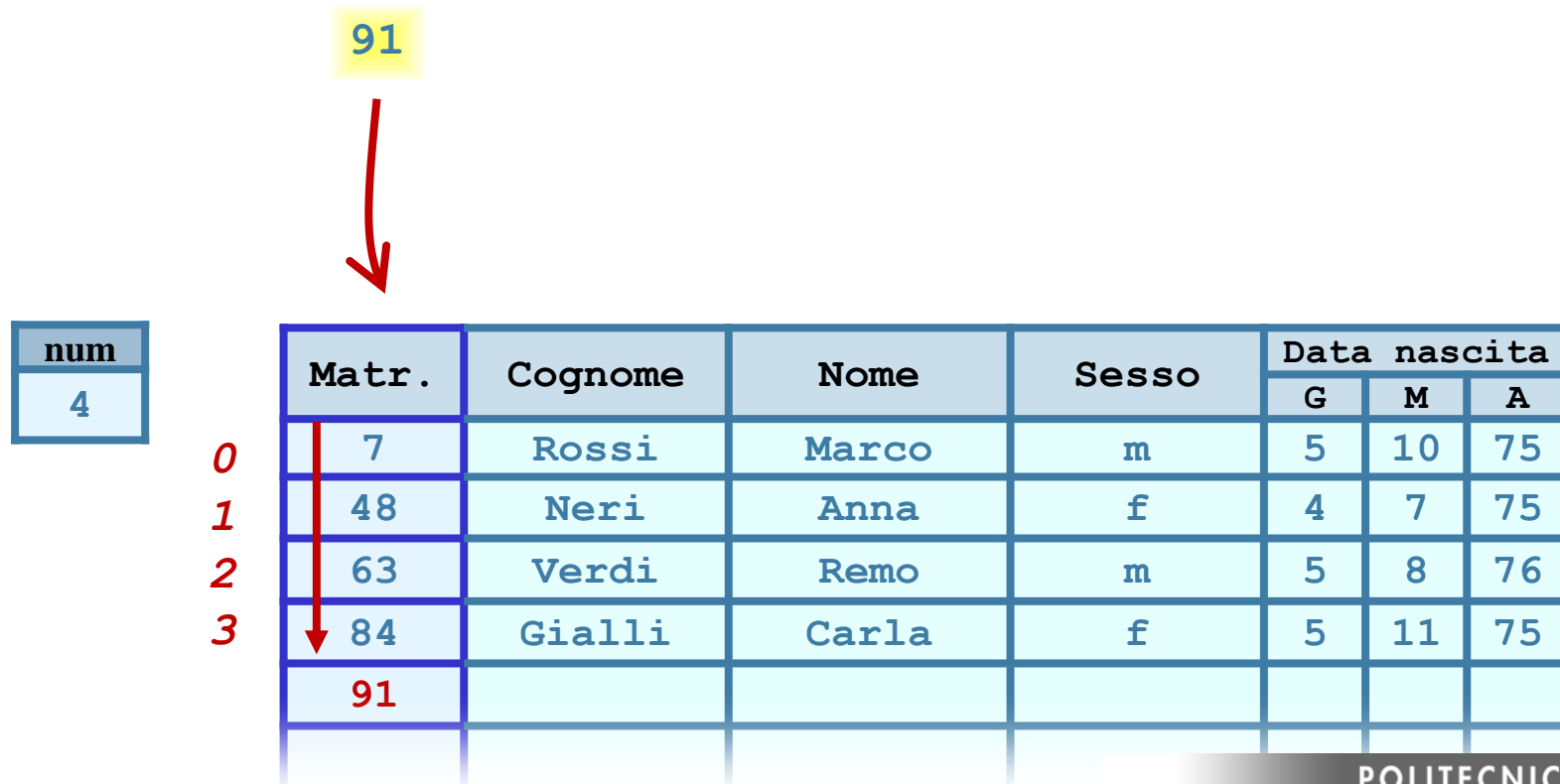
0
1
2
3

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
						


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

num								
4								

		Matr.	Cognome	Nome	Sesso	Data nascita		
						G	M	A
0		7	Rossi	Marco	m	5	10	75
1		48	Neri	Anna	f	4	7	75
2		63	Verdi	Remo	m	5	8	76
3		84	Gialli	Carla	f	5	11	75
		91						



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

91



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 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

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

num								
4								

		Matr.	Cognome	Nome	Sesso	Data nascita		
						G	M	A
0	↓	7	Rossi	Marco	m	5	10	75
1		48	Neri	Anna	f	4	7	75
2		63	Verdi	Remo	m	5	8	76
3		84	Gialli	Carla	f	5	11	75

50



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

50

num
4

	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
0	7	Rossi	Marco	m	5	10	75
1	48	Neri	Anna	f	4	7	75
2	63	Verdi	Remo	m	5	8	76
3	84	Gialli	Carla	f	5	11	75

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

50



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 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

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

5



num

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 inserisciSeNonEsiste(Studente, InsStudenti &);
```

5

num

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

Convenzione

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



num	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
-1							
0							
1	7	Rossi	Marco	m	5	10	75
2	48	Neri	Anna	f	4	7	75
3	63	Verdi	Remo	m	5	8	76
	84	Gialli	Carla	f	5	11	75

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
int main()
```

```
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

elimina '48'



num	Matr.	Cognome	Nome	Sesso	Data nascita		
					G	M	A
4	7	Rossi	Marco	m	5	10	75
0	48	Neri	Anna	f	4	7	75
1	63	Verdi	Remo	m	5	8	76
2	84	Gialli	Carla	F	5	11	75
3							

ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
int main()
```

```
{...}
```

num

4

'48'

↓

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

POLITECNICO

'48'



ambiente globale

```
InsStudenti inizializzaTabella();
```

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

```
void eliminaSeEsiste(int, InsStudenti &);
```

```
void stampaTabella(InsStudenti &);
```

```
int main()
```

```
{...}
```

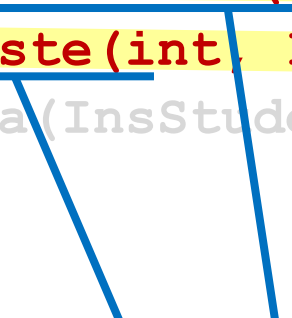


sottoproblema

- esiste un elemento con una certa matricola?
- **se si**, restituisci la matricola dell'elemento precedente
- **se no**, restituisci la matricola dell'elemento dopo il quale effettuare l'inserimento

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

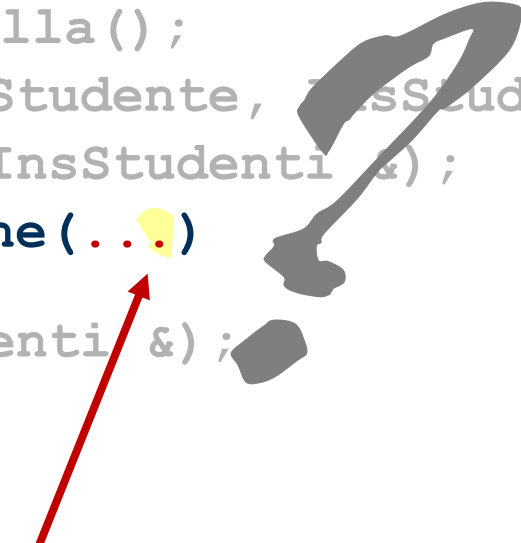


cercaSeEsisteEPosizione

- esiste un elemento con una certa matricola?
- **se si**, restituisci la matricola dell'elemento precedente
- **se no**, restituisci la matricola dell'elemento dopo il quale effettuare l'inserimento

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(...)  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```



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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(..., instStudenti &)  
  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(int, instStudenti &,  
                                ...);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(int, instStudenti &,  
int &, int &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(int, instStudenti &,  
                             int &, int &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(int, instStudenti &,  
int &, int &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

ambiente globale

```
InsStudenti inizializzaTabella();  
void inserisciSeNonEsiste(Studente, InsStudenti &);  
void eliminaSeEsiste(int, InsStudenti &);  
void cercaSeEsisteEPosizione(int, instStudenti &,  
int &, int &);  
void stampaTabella(InsStudenti &);  
  
int main()  
{...}
```

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

Commento

```
void cercaSeEsisteEPosizione(int dato, instStudenti &tab,
                             int &esiste, int &pos)
/* cerca in tab se esiste uno studente con matricola
 * uguale a dato:
 * - se SI (caso utile in eliminazione)
 *   esiste assume valore VERO,
 *   pos individua lo studente precedente.
 *   Se lo studente con matricola uguale a dato è in prima
 *   posizione, pos assume convenzionalmente il valore -1.
 * - se NO (caso utile in inserimento)
 *   esiste assume valore FALSO,
 *   pos individua lo studente dopo il quale deve essere
 *   inserimento il nuovo studente per mantenere l'ordinamento.
 *   Se il nuovo studente deve essere inserito in prima
 *   posizione pos assume convenzionalmente il valore -1.
 */
{
    ...
}
```


Passaggio per indirizzo

```
void cercaSeEsisteEPosizione(int dato, instStudenti &tab,
                             int &esiste, int &pos)
/* cerca in tab se esiste uno studente con matricola
 * uguale a dato:
 * - se SI (caso utile in eliminazione)
 *   esiste assume valore VERO,
 *   pos individua lo studente precedente.
 *   Se lo studente con matricola uguale a dato è in prima
 *   posizione, pos assume convenzionalmente il valore -1.
 * - se NO (caso utile in inserimento)
 *   esiste assume valore FALSO,
 *   pos individua lo studente dopo il quale deve essere
 *   inserimento il nuovo studente per mantenere l'ordinamento.
 *   Se il nuovo studente deve essere inserito in prima
 *   posizione pos assume convenzionalmente il valore -1.
 */
{
    ...
}
```

```
void cercaSeEsisteEPosizione(int dato, instStudenti &tab,  
                             int &esiste, int &pos)
```

```
/* cerca in tab se esiste uno studente con matricola  
 * uguale a dato:  
 * - se SI (caso utile in eliminazione)  
 *   esiste assume valore VERO,  
 *   pos individua lo studente precedente.  
 *   Se lo studente con matricola uguale a dato è in prima  
 *   posizione, pos assume convenzionalmente il valore -1.  
 * - se NO (caso utile in inserimento)  
 *   esiste assume valore FALSO,  
 *   pos individua lo studente dopo il quale deve essere  
 *   inserimento il nuovo studente per mantenere l'ordinamento.  
 *   Se il nuovo studente deve essere inserito in prima  
 *   posizione pos assume convenzionalmente il valore -1.  
 */
```

```
{
```

```
...
```

```
}
```

C++

```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                            esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
      << setw(5) << nuovoStudente.matricola << endl;
      int temp = posizione + 1;
      for (int i = tab.num - 1; i >= temp; i--)
          tab.elenco[i + 1] = tab.elenco[i];
      tab.elenco[temp] = nuovoStudente;
      tab.num++;
    }
}
```

Si possono effettuare dichiarazioni
anche nel corpo della funzione...
... a patto che la dichiarazione
preceda l'uso

```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                             esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
      << setw(5) << nuovoStudente.matricola << endl;
      int temp = posizione + 1;
      for (int i = tab.num - 1; i >= temp; i--)
          tab.elenco[i + 1] = tab.elenco[i];
      tab.elenco[temp] = nuovoStudente;
    }
```

num
4

0
1
2
3

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



```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                             esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
        << setw(5) << nuovoStudente.matricola << endl;
        int temp = posizione + 1;
        for (int i = tab.num - 1; i >= temp; i--)
            tab.elenco[i + 1] = tab.elenco[i];
        tab.elenco[temp] = nuovoStudente;
        tab.num++;
    }
}
```

```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                             esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
        << setw(5) << nuovoStudente.matricola << endl;
        int temp = posizione + 1;
        for (int i = tab.num - 1; i >= temp; i--)
            tab.elenco[i + 1] = tab.elenco[i];
        tab.elenco[temp] = nuovoStudente;
        tab.num++;
    }
}
```

```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                             esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
        << setw(5) << nuovoStudente.matricola << endl;
        int temp = posizione + 1;
        for (int i = tab.num - 1; i >= temp; i--)
            tab.elenco[i + 1] = tab.elenco[i];
        nuovoStudente = nuovoStudente;
        tab.num++;
    }
}
```

```
void inserisciSeNonEsiste(Studente nuovoStudente,
                          InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(nuovoStudente.matricola, tab,
                             esiste, posizione);
    if ((esiste == FALSO) && (tab.num <= dim))
    { cout << "inserimento di studente con matricola"
        << setw(5) << nuovoStudente.matricola << endl;
        int temp = posizione + 1;
        for (int i = tab.num - 1; i >= temp; i--)
            tab.elenco[i + 1] = tab.elenco[i];
        tab.elenco[temp] = nuovoStudente;
        tab.num++;
    }
}
```



```
void eliminaSeEsiste(int matricola, InsStudente &tab)
{
    int esiste, posizione;
    cercaSeEsisteEPosizione(matricola, tab, esiste, posizione);
    if ((esiste == VERO)
        { cout << "eliminazione di studente con matricola"
            << setw(5) << matricola << endl;
            int temp = posizione + 1;
            for (int i = temp + 1; i < tab.num; i++)
                tab.elenco[i - 1] = tab.elenco[i];
            tab.num--;
        }
}
```