



## 1. Ricerca di un elemento in un array ordinato (ricerca dicotomica)

La ricerca di un elemento all'interno di un array ordinato può essere realizzata in modi diversi rispetto alla ricerca sequenziale: uno dei più noti è quello detto della **ricerca dicotomica**.

La ricerca dicotomica presuppone che gli elementi del vettore siano **ordinati**. Si usa il termine *dicotomica* perché si procede a divisioni successive dell'array in cui si effettua la ricerca, scartando le parti che non possono contenere il termine cercato.

Se scartiamo la parte a sinistra, la metà diventa il nuovo estremo sinistro della ricerca; se scartiamo la parte a destra, la metà diventa il nuovo estremo destro della ricerca. La divisione a metà dell'intervallo di ricerca continua finché l'elemento viene trovato e in questo caso si ottiene come risultato la posizione dell'elemento nell'array, oppure finché l'intervallo di ricerca si riduce a due elementi consecutivi dell'array e l'elemento da ricercare risulta mancante nell'array.

### PROGETTO

#### Cercare un elemento in un array ordinato e comunicare il risultato della ricerca.

PROGRAMMA C++

```
// RicerDic.cpp: ricerca in un array ordinato
#include <iostream>
using namespace std;

    const int MAX = 100;
    // variabili globali
    int n;        // numero da ricercare

// prototipi delle funzioni
int ChiediDimensione();
void CaricaVettore(int v[], int d);
void Ordina(int v[], int d);
void Scambia(int& a, int& b);
bool Ricerca(int v[], int d);

// funzione principale
int main()
{
    int dim;           // dimensione indicata dall'utente
    int vett[MAX];    // vettore da ordinare
    bool trovato;     // indicatore se trovato

    dim = ChiediDimensione();
    cout << "Carica gli elementi del vettore" << endl;
    CaricaVettore(vett,dim);
    Ordina(vett,dim);
    cout << "Numero da cercare: ";
    cin >> n;
    trovato = Ricerca(vett, dim);
    if (trovato) cout << n << " e' presente nel vettore" << endl;
    else cout << n << " non e' presente nel vettore" << endl;

    return 0;
}
```



```

// dimensione dell'array
int ChiediDimensione()
{
    int d;
    do {
        cout << "Dimensione dell'array: ";
        cin >> d;
    } while (d<1 || d>MAX);

    return d;
} // ChiediDimensione

// caricamento delle componenti
void CaricaVettore(int v[], int d)
{
    for (int i=0; i<d; i++) {
        cout << "Elemento di posto " << i << ": ";
        cin >> v[i];
    }
} // CaricaVettore

// ordinamento crescente dell'array
void Ordina(int v[], int d)
{
    for (int i=0; i<d-1; i++) {
        for (int j=i+1; j<d; j++)
            if (v[i]>v[j]) Scambia(v[i], v[j]);
    }
} // Ordina

// scambio di due variabili
void Scambia(int& a, int& b)
{
    int comodo;

    comodo = a;
    a = b;
    b = comodo;
} // Scambia

// ricerca dell'elemento nell'array
bool Ricerca (int v[], int d)
{
    bool s = false;    // esito della ricerca
    int sx;            // estremo sinistro della ricerca
    int dx;            // estremo destro della ricerca
    int md;            // punto medio dell'intervallo

    sx = 0;
    dx = d-1;
    do {
        md = (sx+dx)/2;
        if (v[sx]==n || v[dx]==n || v[md]==n) s = true;
        else
            if (v[md]<n) sx = md + 1;
            else dx = md - 1;
    } while (sx<=dx && s==false);

    return s;
} // Ricerca

```



## ESERCIZI

- 1 Dato un elenco di matricole di studenti organizzate in un array, dopo aver ordinato in modo crescente l'array, controllare se il valore di una matricola richiesta da tastiera è compresa nell'elenco.
- 2 Gli importi delle fatture emesse nel mese di marzo sono organizzati in un array. Dopo aver ordinato in modo crescente l'array:
  - visualizzare l'importo massimo (importo corrispondente all'indice più alto dell'array).
  - controllare se il valore di un importo fornito da tastiera è compreso nell'array (supponendo che gli importi siano tutti diversi tra loro).