

Algoritmo per la ricerca sequenziale di un array

Il metodo di **ricerca di un elemento in un array** consiste nell'esaminare le componenti dell'array in sequenza, vale a dire una dopo l'altra a partire dalla componente con il primo indice, secondo l'ordine che esse occupano nell'array, per controllare se tra le componenti dell'array ne esiste una avente valore uguale a quello cercato.

Ci sono vari tipi di ricerca, a seconda del tipo di problema che occorre risolvere. In alcuni casi è sufficiente sapere se un elemento è presente o meno in un vettore, senza preoccuparsi della sua posizione, in altri casi è molto utile sapere in che posizione è situato l'elemento cercato. Oppure interessa solamente sapere se l'elemento ricercato è presente più volte, oppure si cercano gli indici di tutte le occorrenze del valore cercato, e così via.

Quando nella ricerca vengono esaminate tutte le componenti dell'array, si dice che viene fatta una **ricerca completa**.

La ricerca si arresta quando l'indice che controlla gli elementi dell'array è arrivato all'ultimo elemento oppure quando è stato trovato un elemento uguale a quello cercato.

Questo metodo può essere efficace per un piccolo numero di componenti, perde di efficienza all'aumentare del numero degli elementi dell'array.

Progetto

Ricerca un elemento all'interno di un array.

Il programma è in grado di ricercare un elemento all'interno di un vettore. Il vettore viene caricato utilizzando le stesse procedure del Progetto 1. Il programma termina visualizzando l'indice al quale si trova l'elemento ricercato oppure il messaggio di valore non trovato.

Dati di input:

vettore di stringhe
valore da ricercare

Dati di output:

posizione dove si trova il valore cercato
oppure
messaggio di non trovato.

Nome del progetto

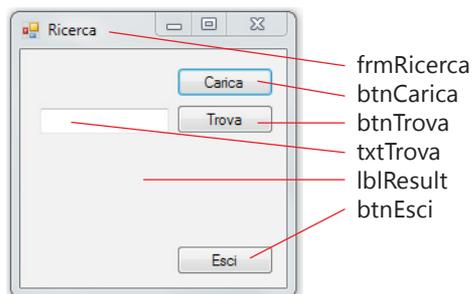
RicercaSequenziale di tipo *Applicazione Windows Form*

Disegno dell'interfaccia grafica

Il form contiene un pulsante di comando *Carica* per inserire i dati nel vettore, una casella di testo per acquisire il dato da cercare e un pulsante di comando *Trova* per avviare la ricerca. Il pulsante *Esci* provoca l'uscita dal programma.

Il programma utilizza anche una finestra di dialogo (*InputDialog*) per acquisire la dimensione dell'array e una *Label* per visualizzare il risultato, che può essere la posizione dell'elemento trovato oppure un messaggio di elemento non trovato.

Classe	Proprietà dell'oggetto	
Form	Name	frmRicerca
	Text	<i>Ricerca</i>
TextBox	Name	txtTrova
Label	Name	lblResult
	Text	
Button	Name	btnCarica
	Text	<i>Carica</i>
Button	Name	btnTrova
	Text	<i>Trova</i>
Button	Name	btnEsci
	Text	<i>Esci</i>



Gestione degli eventi

Il vettore viene caricato come nei progetti precedenti. Dopo aver caricato il vettore, viene inserito il testo da cercare nella casella di ricerca: facendo clic sul pulsante *Trova*, si avvia la procedura di ricerca. L'algoritmo di ricerca verifica se è presente il termine da ricercare fornito nella casella di testo: assegna il valore falso alla variabile booleana *trovato* e inizia la ricerca confrontando tutti gli elementi del vettore con la stringa cercata, attraverso una ripetizione precondizionale. In caso di esito negativo per la ricerca avvisa l'utente.

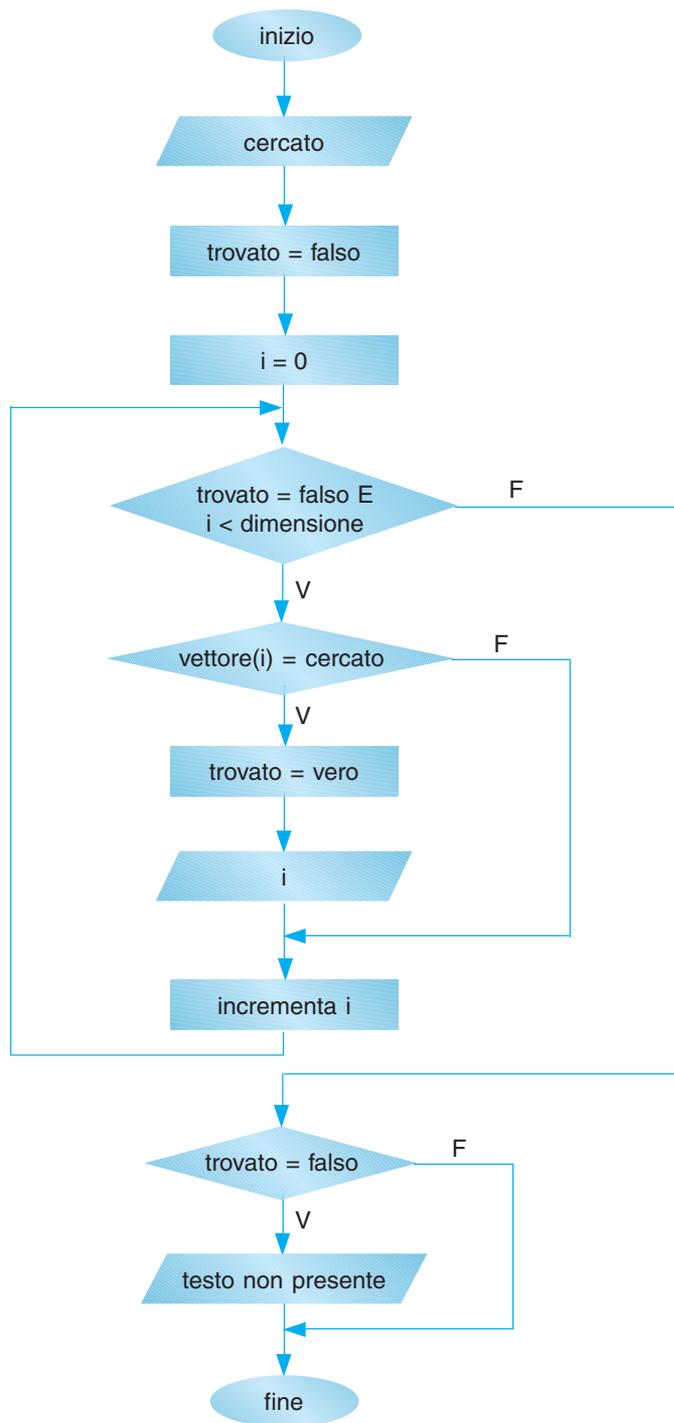
Algoritmo in pseudocodifica

```

inizio
  immetti cercato
  assegna trovato = falso
  assegna i = 0
  esegui mentre trovato = falso E i < dimensione
    se vettore(i) = cercato
      allora
        assegna trovato = vero
        scrivi indice i
      fine se
    incrementa i
  ripeti
  se trovato = falso
  allora
    scrivi "il testo non è presente"
  fine se
fine

```

Diagramma a blocchi



Codice Visual Basic

```
Const Max As Integer = 10
Dim dimensione As Integer = 0
Dim vettore() As String
Dim trovato As Boolean
Dim i As Integer

Private Sub btnCarica_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCarica.Click
    Do
        dimensione = InputBox("Inserisci la dimensione", "Chiedi dimensione")
    Loop Until dimensione >= 1 And dimensione <= Max
    ReDim vettore(dimensione - 1)
    For i = 0 To dimensione - 1
        vettore(i) = InputBox("Inserisci il " & i + 1 & "° elemento",
        "Inserisci")
    Next
End Sub

Private Sub btnTrova_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnTrova.Click
    trovato = False
    i = 0
    Dim cercato As String
    If txtTrova.Text = "" Then
        MessageBox.Show("Inserire il testo da ricercare", "Attenzione")
    Else
        cercato = txtTrova.Text
        Do While trovato = False And i < dimensione
            If vettore(i) = cercato Then
                trovato = True
                lblResult.Text = "Il valore si trova all'indice " & i
            End If
            i += 1
        Loop
        If trovato = False Then
            lblResult.Text = "Il testo " & cercato & " non è presente"
        End If
    End If

End Sub

Private Sub btnEsci_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEsci.Click
    End
End Sub
```

Non è possibile stabilire a priori il numero di passi necessari per la ricerca. Se il valore cercato si trova all'inizio, l'algoritmo termina dopo una sola iterazione.