

RICERCA DI UN ELEMENTO

- Si legga da tastiera un array di N elementi (N stabilito dall'utente)
- Si richieda un elemento x
- il programma deve cercare l'elemento x nell'array
- Se l'elemento è presente, deve visualizzare la posizione in cui si trova l'elemento

Primo livello di specifica

- Leggi N
- Leggi l'array di N elementi
- Leggi x
- Per tutti gli elementi dell'array (con indice i da 0 a $N-1$)
 - Confronta x con l'elemento di indice i .
Se x è uguale all'elemento i -esimo, visualizza i
 - altrimenti continua la ricerca

ESEMPIO: Ricerca di un elemento

```
#include <stdio.h>
#define N 15
main ()
{int v, pos, n, i, a[N];
  printf("Inserisci N:"); scanf("%d",&n);
  for(i=0;i<n;i++)
    scanf("%d",&a[i]);
  printf ("Valore da cercare: ");
  scanf ("%d",&v);
  pos=0;
  while (pos<n && a[pos]!=v)
    pos++;
  if (pos==n) printf("Non trovato\n");
  else printf("Trovato in pos %d",pos);
}
```

Ricerca con sentinella

- Nel ciclo di una ricerca dobbiamo inserire due condizioni:
 - finché l'array non è finito
 - finché non ho trovato l'elemento
- In alcuni casi si potrebbe, in alternativa, aggiungere l'elemento da cercare alla fine dell'array: in questo modo la condizione del ciclo risulta semplificata



Ricerca con sentinella

```
#include <stdio.h>
#define N 15
main ()
{int v, pos, n, i, a[N];
 printf("Inserisci N:"); scanf("%d",&n);
 for(i=0;i<n;i++)
   scanf("%d",&a[i]);
 printf ("Valore da cercare: ");
 scanf ("%d",&v);
 a[n]=v;
 pos=0;
 while (a[pos]!=v)
   pos++;
 if (pos==n) printf("Non trovato\n");
   else printf("Trovato in pos %d",pos);
}
```

Sentinella: inserisco il valore da cercare in fondo all'array

Nella condizione del while non ho più bisogno di controllare se sono arrivato in fondo

ORDINAMENTO DI UN VETTORE

- Dati n valori interi forniti in ordine qualunque, stampare in uscita l'elenco dei valori dati in ordine crescente.
- Vettore di elementi di un certo tipo, sul quale è definita una **relazione d'ordine totale** (ad esempio, tipo float)
 - **Naive sort (o selection sort, o ordinamento per minimi successivi):**
 - Ad ogni passo seleziona il minimo nel vettore e lo pone nella prima posizione

ESEMPIO: Ordinamento di un vettore

- Secondo livello di specifica per l'ordinamento
while (<il vettore ha più di una componente>
 { <individua il minimo nel vettore corrente in
 posizione **posmin**>
 <scambia se necessario il primo elemento
 del vettore corrente con quello in posizione **posmin** >
 <considera come vettore corrente quello
 precedente tolto il primo elemento>
 }

Naive sort – programmazione Top-Down

```
main()
{ int v[N], n, i, posmin;
  // lettura di V, n
  for (i=0; i<n-1; i++)
  {
    /* cerca il minimo del vettore da i+1 a
    n: posmin */
    if (v[posmin]<v[i])
      // scambia v[posmin] e v[i]
  }
  // stampa v
}
```

Naive sort – programmazione Top-Down

```
main()
{ int v[N], n, i, posmin;
  // lettura di V, n
  for (i=0; i<n-1; i++)
  {
    /* cerca il minimo del vettore da i+1 a n: posmin */
    int j;
    posmin=i+1;
    for(j=i+2; j<n; j++)
      if (v[posmin]>v[j])
        posmin=j;

    if (v[posmin]<v[i])
      // scambia v[posmin] e v[i]
  }
  // stampa v
}
```

Naive sort – programmazione Top-Down

```
main()
{ int v[N], n, i, posmin;
  // lettura di V, n
  for (i=0; i<n-1; i++)
  {
    /* cerca il minimo del vettore da i+1 a n: posmin */
    int j;
    posmin=i+1;
    for(j=i+2; j<n; j++)
      if (v[posmin]>v[j])
        posmin=j;

    if (v[posmin]<v[i])
    { // scambia v[posmin] e v[i]
      int app;
      app=v[posmin];
      v[posmin]=v[i];
      v[i]=app;
    }
  }
  // stampa v
}
```

OSSERVAZIONI – Naive Sort

- Si eseguono tutti i confronti anche se il vettore è già ordinato (esistono altri metodi più efficienti, li vedrete in altri corsi).

Esempio: Array di struct

Scrivere un programma C che definisca la struttura “esame” composta dal nome dell’esame (stringa) e dal voto (intero).

Si leggano poi da terminale n esami, con n definito dall’utente (max 30), e si inseriscano in un array.

L’utente inserisca poi il nome di un esame da cercare.

Si utilizzi una ricerca con sentinella per trovare tale esame nell’array (e lo si fornisca in uscita).

Suggerimento 1: Non usate spazi nel nome dell’esame.

Suggerimento 2: Utilizzare le funzioni **strcpy** e **strcmp** contenute nella libreria **string.h**

Array di struct – soluzione (1/2)

```
#include <stdio.h>
#include <string.h>
#define NMAX 30
typedef struct
{   char nome[20];
    int voto;
} esame;

void main() {
    esame esamimarco[NMAX];
    char cerca[20];
    int n, i, pos;

    printf("Inserisci il numero di esami:");
    scanf("\n%d", &n);

    //inserimento dati
    for (i=0; i<n; i++) {
        printf("\nInserisci il nome dell'esame:");
        scanf("\n%s", esamimarco[i].nome);
        printf("Inserisci il voto:");
        scanf("\n%d", &esamimarco[i].voto);}
```

Continua...

Array di struct – soluzione (2/2)

```
printf("\nInserisci il nome dell'esame da cercare:");
scanf("\n%s", cerca);

pos=0; //inizializzo contatore

strcpy(esamimarco[n].nome, cerca); //sentinella (solo il
//campo su cui si effettua la ricerca)

//ricerca
while(strcmp(esamimarco[pos].nome, cerca)) pos++;

if (pos==n) {
    printf("\nEsame non trovato.\n");
} else {
    printf("\nTrovato in posizione %d.\n%s\t%d\n", pos+1,
esamimarco[pos].nome, esamiarco[pos].voto);
}
}
```

ESERCIZI PROPOSTI

Esercizio 1 - Struct

Scrivere un programma C che definisca la struttura “misuraUK” composta da quattro interi (miglia, yarde, piedi, pollici), e una struttura “misuraUE” composta da tre interi (km, metri, mm). Si richieda poi una misura inglese e si stampi a video la misura europea.

Ricordate che:

1 miglio = 1760 yarde

1 yarda = 3 piedi

1 piede = 12 pollici

1 pollice = 25,4 mm

ESERCIZI PROPOSTI

Esercizio 2 - Struct

Scrivere un programma C che definisca la struttura “persona” composta da nome, cognome, indirizzo, provincia e data di nascita. Si richiedano tutti i dati e si stampino a video.

ESERCIZI PROPOSTI

Esercizio 3 - Array

Si scriva un programma che richieda in ingresso un numero di elementi N (<20) e poi N interi, inserendoli in un array. Chieda poi una soglia S e si copino tutti gli elementi del vettore in altri due vettori, separando elementi sopra la soglia e sotto la soglia.

Esercizio 4 - Array

Si scriva un programma C che chieda in ingresso due matrici 3×3 e poi chieda all'utente cosa vuole fare: somma, sottrazione o moltiplicazione. Si stampi a video la matrice risultato.

ESERCIZI PROPOSTI

Esercizio 5 - Array

Si scriva un programma che richieda in ingresso l'andamento annuale del valore delle azioni (12 valori, uno per ogni mese) e crei un vettore. Poi stampi a video un istogramma orizzontale:

```
1  ***** 110
2  ***** 90
3  ***** 140
4  ***** 160
5  ***** 150
...

```

N.B. Visualizzare al massimo 50 stelle

ESERCIZI PROPOSTI

Esercizio 6 - Array di struct

Si realizzi un programma C che legga da utente i dati relativi ad alcuni corsi. In particolare, per ogni corso vengono dati:

- **denominazione** del corso: una stringa di 20 caratteri che riporta il nome del corso;
- **cognome** del docente: una stringa di 15 caratteri che rappresenta il cognome del docente del corso;
- **iscritti**: un intero che indica il numero di studenti che frequentano il corso.

Il programma deve stampare la denominazione del corso e il cognome del docente relativi a tutti i corsi che hanno il numero di iscritti maggiore o uguale alla media aritmetica degli iscritti (calcolata su tutti i corsi).

ESERCIZI PROPOSTI

Esercizio 7 - Array di struct

• Si scriva un programma C che legga una serie di dati e li memorizzi in un vettore SQUADRE (di dimensione 3) contenente strutture (**struct squadra**) del tipo:

- **nome squadra** (*stringa di lunghezza 20*)

- **codice squadra** (*intero*)

- **goal fatti** (*intero*)

- **goal subiti** (*intero*)

• Stampi a terminale tutti i nomi e codici delle squadre che hanno fatto un numero di goal maggiore del numero dei goal subiti.

• Letto a terminale un codice di una squadra stampi a video il nome della squadra, i goal fatti e i goal subiti.

ESERCIZI PROPOSTI

Attenzione: abbiamo bisogno di un ARRAY di strutture !!!!!

Esempio: l'utente inserisce i seguenti dati per 3 squadre

juventus

1

10

12

milan

2

7

6

inter

3

13

11

<i>juventus</i>	<i>milan</i>	<i>inter</i>
1	2	3
10	7	13
12	6	11

- Viene stampato a video

milan **2**

inter **3**

- Se l'utente digita 1 viene stampato

juventus **10 12**

ESERCIZI PROPOSTI

Esercizio 8 - Array di Struct

Scrivere un programma C che crei una rubrica telefonica, usando una struct "persona" (nome, cognome, numero di telefono, e-mail).

Il programma chiede di inserire delle nuove persone, poi si mette in modalità ricerca e, chiesto il nome stampa a video la scheda corrispondente.

ESERCIZI IN LABORATORIO

Esercizio 1 - Struct

Scrivere un programma C che definisca la struttura “misuraUK” composta da quattro interi (miglia, yard, piedi, pollici), e una struttura “misuraUE” composta da tre interi (km, metri, mm). Si richieda poi una misura inglese e si stampi a video la misura europea.

Esercizio 5 - Array

Si scriva un programma che richieda in ingresso l’andamento annuale del valore delle azioni (12 valori, uno per ogni mese) e crei un vettore. Poi stampi a video un istogramma orizzontale

Esercizio 8 - Array di Struct

Scrivere un programma C che crei una rubrica telefonica, usando una struct “persona” (nome, cognome, numero di telefono, e-mail).

Il programma chiede di inserire delle nuove persone, poi si mette in modalità ricerca e, chiesto il nome stampa a video la scheda corrispondente.