ELEMENTI DI INFORMATICA L-B

Ing. Claudia Chiusoli

Materiale

- Lucidi delle lezioni
- Date degli appelli
- Testi di esami precedenti
- Informazioni e contatti

http://www.lia.deis.unibo.it/Courses/

Programma del corso

- Elementi di programmazione: C

- I sistemi informativi
  - Base di dati
  - Schema E-R
  - Linguaggio SQL
  - Base di dati sulla rete internet

Fondamenti di C
Tipi di dato (scalari)

- Caratteri:
  - **char** (caratteri ascii)
- Interi
  - Con segno
    - **short**: *16 bit -32.768 a +32.768*
    - **Int** *dipende dal compilatore*
    - **long**: *32 bit*
  - Senza segno (Naturali)
    - **unsigned short**: *16 bit da 0 a …*
    - **unsigned**
    - **unsigned long**: *32 bit da 0 a …*
- Reali
  - **float** (singola precisione 32 bit) da 10^{-38} a 10^{38}
  - **double** (doppia precisione 64 bit) da 10^{-308} a 10^{308}

Fondamenti di C
Tipi interessanti

- Boolean:
  - **zero**: indica **falso**
  - Tutto il resto: indica **vero** (utilizzare **uno**)

- Stringhe: sequenze di caratteri

  *char *stringa;*

  char stringa[10];

  *typedef char * Stringa;*

Fondamenti di C
Input / Output

#include <stdio.h>

scanf(<stringa_formato>, <sequenza_variabili>)
printf(<stringa_formato>, <sequenza_elementi>)

int x;
float y;
scanf("%d%f", &x, &y);
printf("%d + %f = %f", x, y, x+y);

Alcuni Formati
%d → int
%f → float
%c → carattere
%s → stringa

Un Semplice Programma

- **Questo programma stampa la tabella di conversione dei gradi Fahrenheit-Celsius**

- **Algoritmo:**

  Definire 2 costante per il massimo e minimo della tabella

  Assegnare un valore di intervallo di gradi tra le singole righe della tabella (step)

  Finch'è non si raggiunge il valore massimo

  calcolare la conversione in celsius

  scrivere i gradi fahrenheit e i rispettivi celsius

  incrementare di uno step i gradi fahrenheit

  Fine.

Un Semplice Programma

Questo programma stampa la tabella di conversione dei gradi Fahrenheit-Celsius

```
#include <stdio.h>
#define MIN 0              // limite inferiore della tabella
#define MAX 300            // limite superiore


int main () {
        int fahr, celsius, step;
        step = 20;              // scanf( “%d”,&step);
        fahr = MIN;
        while (fahr <= MAX) {
                celsius = (fahr-32) * 5 / 9;
                printf ("%d\t%d\n", fahr, celsius);
                fahr += step;
        }
        return 0;
 }
```

Array

- Una collezione **finita** di N variabili dello **stesso tipo**, identificabili con un indice compreso tra **0** e **N-1**

- **Definizione:**

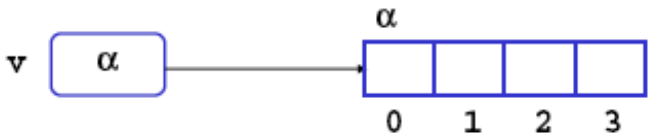
  <tipo> <nome array> [<COSTANTE>];

- **Esempi:**

  int vettore [10]; char stringa[255];

Array

- **L'array è un puntatore** ad un'area di memoria pre-allocata, che **contiene l'indirizzo del primo elemento** del vettore.



- Indicano tutti la stessa cosa

  v &v[0] α

  v+1 &v[1] α+1

Array

- Per assegnare un array: eseguire un ciclo su ogni elemento

- Passaggio di array come parametri alle funzioni: **sempre per riferimento!**

int v[10];

…

ordina (v); → *v = l'indirizzo dell'area di memoria dell'array*

Esercizio - Calcolo della media

- Calcolare la media di N interi di un vettore v assegnato tramite std input

- Algoritmo:
  - Definire una costante N
  - Per N volte, leggere da input un intero
  - Scrivere la media

Esercizio - Calcolo della media

- Calcolare la media di N interi di un vettore v assegnato tramite std input

```
#include <stdio.h>
#define N 10
double media (int v[]);      //prototipo della funzione = dichiarazione

int main() {
      int v[N], i;
      for(i=0;i<N;i++)   scanf("%d",&v[i]);
      printf("La media vale %f",media(v));
}

double media(int vet[]){
      int somma=0,i;
      for(i=0;i<N;i++) somma+=vet[i];
      return somma/N;
}
```

Esercizio - Ricerca sequenziale

- Scrivere la funzione per la ricerca **sequenziale** di un elemento in un array
- Il numero massimo di confronti è la dimensione dell'array (n), e si verifica quando l'elemento che stiamo cercando è in v[n-1].
- Algoritmo:
  - Per ogni elemento dell'array
  - Se l'elemento i-esimo coincide con l'elemento da cercare, terminare l'esecuzione della funzione e restituire l'indice i
  - Se non si è trovato l'elemento in nessuna posizione restituire -1

```
int  ricercaSequenziale(int vet[], int dim, int el){
     int i;
     for(i=0;i<dim;i++)
       if (vet[i]==el)
             return i;
     return -1;
}
```

Esercizio - Ricerca binaria

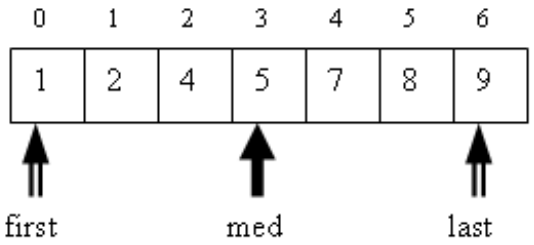
- La ricerca binaria si applica su array ORDINATI
- Confronta l'elemento centrale dell'array e restringe la successiva iterazione alla metà che può contenere la chiave cercata

1° ciclo:

first=0;

last=n-1;

med=(last+first)/2;



if (elemento = = v[med]) return med;

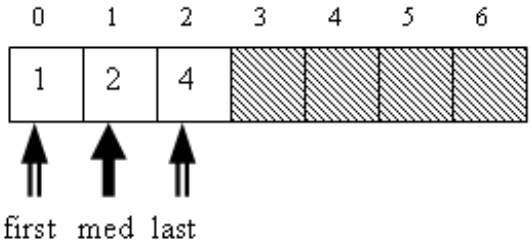
2° ciclo:

if (elemento < v[med])

last= med-1;

else first = med +1;

med=(last+first)/2;



Esercizio - Ricerca binaria

- Esempio: dato un array di 1023 elementi è possibile restringere la ricerca a 511 di essi con un solo confronto. Un altro confronto e la ricerca sarà fra 255 valori. Di fatto si può cercare in tutto l'array con solo 10 confronti.

- Scrivere la funzione per la **ricerca binaria** di un elemento in un array **ORDINATO**

  Algoritmo

  – Inizializzare 3 variabili intere: first al primo indice dell'array, last all'ultimo e med al medio tra first e last
  – Inizializzare la variabile booleana *trovato* a false

  – Finché first <= last e non è stato trovato l'elemento da cercare
    - se la posizione med-esima contiene l'elemento cercato allora trovato = true
    - altrimenti scarto la metà degli elementi del vettore troppo grandi o troppo piccoli

  – Restituire l'indice in cui si trova l'elemento oppure -1 se l'array non lo contiene

Esercizio - Ricerca binaria

- Scrivere la funzione per la **ricerca binaria** di un elemento in un array **ORDINATO**

```
int  ricercaBinaria(int vet[], int n, int el){

     int first=0, last=n-1, med=(first+last) / 2;
     int trovato=0;     //boolean → false

     while ( (first<=last) && (trovato==0) ){

       if (el == vet[med])  trovato=1;
       else {
            if (el < vet[med] ) last = med -1;
            else first = med +1;
            med = (firs + last) / 2;
       }
     }

     if (trovato==1) return med;
     else return -1;
}
```