

LIA - DEIS - Facoltà di Ingegneria - Università di Bologna
Elementi di informatica L

Corso di laurea in Ingegneria Informatica - Anno Accademico 2007-2008

Progetti su più File Array e Funzioni

ESERCIZIO 5.1

Scrivere un modulo in C, `vet_io.c` che:

- contenga una procedura `leggi_ivet` che legge da tastiera un vettore di interi
- contenga una procedura `stampa_ivet` che stampa su video un vettore di interi
- contenga una procedura `leggi_dvet` che legge da tastiera un vettore di double
- contenga una procedura `stampa_dvet` che stampa su video un vettore di double

Scrivere inoltre un main di prova che legge un vettore di interi `lambda[]` ed uno di reali `v[]` e li visualizza. Deve inoltre definire un terzo vettore di reali, `alfa[]`, i cui elementi sono il prodotto degli elementi omologhi dei vettori `lambda[]` e `v[]`. Il programma visualizzi infine il vettore `alfa[]`. Si strutturi il programma come progetto costituito da moduli su più file.

Soluzione:

Il progetto si compone di tre file:

- `vet_io.c`: file che contiene la definizione delle procedure
- `vet_io.h`: header file del modulo `vet_io.c`
- `scalare.c`: file contenente il main.

`vet_io.c`

```
#include <stdio.h>

void leggi_ivet(int v[], int n)
{
    int i;
```

```
for (i=0; i<n; i++) {
    printf("v[%d] = ",i);
    scanf("%d",&v[i]);
}
}

void stampa_ivet(int v[], int n)
{
    int i;
    for (i=0; i<n; i++) printf("%d ",v[i]);
    printf("\n");
}

void leggi_dvet(double v[], int n)
{
    int i;
    for (i=0; i<n; i++) {
        printf("v[%d] = ",i);
        scanf("%lf",&v[i]);
    }
}

void stampa_dvet(double v[], int n)
{
    int i;
    for (i=0; i<n; i++) printf("%lf ",v[i]);
    printf("\n");
}
```

vet_io.h

```
void leggi_ivet(int v[], int n);

void stampa_ivet(int v[], int n);

void leggi_dvet(double v[], int n);

void stampa_dvet(double v[], int n);
```

scalare.c

LIA

```
#include "vet_io.h" /* vet_io.h e' nella stessa directory del main
*/

#define N 5

main()
{
    int lambda[N];
    double v[N],alfa[N];
    int i;

    leggi_ivet(lambda,N);
    stampa_ivet(lambda,N);
    leggi_dvet(v,N);
    stampa_dvet(v,N);

    for (i=0; i<N; i++) alfa[i] = lambda[i]*v[i];

    stampa_dvet(alfa,N);
}
```

ESERCIZIO 5.2

Si scriva un programma in linguaggio C, che legga una sequenza di caratteri terminata da <return> ('\n') e stampi le frequenze dei caratteri '0','1','2','3','4','5','6','7','8','9'.

```
#include <stdio.h>

#define DIM 10

int main(void)
{
    int i;
    int a[DIM] = {0};
    char ch = ' ';

    printf("Immetti una sequenza di caratteri terminata da '\\n':\n");

    while (ch != '\n') {
        if (ch >= '0' && ch <= '9')
            a[ch - '0']++;
        scanf("%c", &ch);
    }

    printf("Le frequenze sono:\n");
    for (i = 0; i < DIM; i++)
        printf("Freq. di %d: %d\n", i, a[i]);
}
```

```
    return 0;
}
```

VERSIONE CON LO SWITCH

```
#include <stdio.h>

#define DIM 10

int main(void)
{
    int i;
    int a[DIM] = {0};
    char ch = ' ';

    while (ch != '\n') {
        switch(ch) {
            case '0': a[0]++; break;
            case '1': a[1]++; break;
            case '2': a[2]++; break;
            case '3': a[3]++; break;
            case '4': a[4]++; break;
            case '5': a[5]++; break;
            case '6': a[6]++; break;
            case '7': a[7]++; break;
            case '8': a[8]++; break;
            case '9': a[9]++; break;
        }
        scanf("%c",&ch);
    }

    printf("Le frequenze sono:\n");
    for (i = 0; i < DIM; i++)
        printf("Freq. di %d: %d\n", i, a[i]);

    return 0;
}
```

ESERCIZIO 5.3

Si scriva un progetto su più file in linguaggio C che legga due serie di dati e li memorizzi in due vettori di strutture.

Nel primo vettore M (di dimensione 3) vengono memorizzati dati del tipo: <matricola, nome, cognome>. Si noti che la matricola identifica univocamente uno studente e che non ci sono due strutture che contengono lo stesso numero di matricola.

LIA

Nel secondo vettore V (di dimensione 7) vengono memorizzati dati del tipo: <matricola, esame, voto>. Possono esserci più record con lo stesso numero di matricola che denotano diversi esami fatti dallo stesso studente.

Il progetto tramite opportune procedure legga i dati in ingresso e li inserisca nei due vettori.

Successivamente per ogni studente con matricola X contenuto nel vettore M sommi tutti i suoi voti ottenuti negli esami contenuti nel vettore V.

```
#include <stdio.h>

#define DIMN 3

#define DIMM 7

struct studente{

    int matricola;

    char Nome[20];

    char Cognome[20];

};

struct esami{

    int matricola;

    char esame[20];

    int voto;

};

void leggistud(int n, struct studente V[]);

void leggiesami(int n, struct esami M[]);

int somma(struct studente s, struct esami E[]);

main(){

    int i, S=0;

    struct studente V[DIMN];

    struct esami M[DIMM];

    leggistud(DIMN,V);
```

LIA

```
leggiesami(DIMM,M);  
  
for(i=0; i< DIMN;i++){  
  
    S= somma(V[i], M);  
  
    printf("Studente %s %s\n", V[i].Cognome, V[i].Nome);  
  
    printf("Somma voti: %d\n",S);  
  
}  
  
}
```

```
void leggistud(int n, struct studente Vet[]){  
  
    int i,j;  
  
    for(i=0;i<n;i++){  
  
        printf("Inserisci Matricola, Nome, Cognome\n");  
  
        scanf("%d",&Vet[i].matricola);  
  
        scanf("%s",Vet[i].Nome);  
  
        scanf("%s",Vet[i].Cognome);  
  
    }  
  
}
```

```
void leggiesami(int n, struct esami Vet[]){  
  
    int i;  
  
    for(i=0;i<n;i++){  
  
        printf("Inserisci Matricola, Esame, Voto\n");  
  
        scanf("%d",&Vet[i].matricola);  
  
        scanf("%s",Vet[i].esame);  
  
        scanf("%d",&Vet[i].voto);  
  
    }  
  
}
```

LIA

```
int somma(struct studente s, struct esami E[]){  
    int i;  
  
    int S=0;  
  
    for(i=0;i < DIMM;i++){  
        if (E[i].matricola == s.matricola) S = S + E[i].voto;  
    }  
  
    return S;  
}
```

ESERCIZIO 5.5

Si scriva un progetto su più files in linguaggio C che tramite:

- la funzione `leggi()`, legga a terminale i dati di N studenti (con N definito come la costante 4) costituiti da Nome, Cognome, Voto, e li inserisca in un vettore;
- calcoli con la funzione `media()` la media dei voti;
- stampi a terminale i nominativi di ciascuno studente;
- stampi poi a terminale il voto medio.

Ad esempio, avendo in ingresso:

Alberani Luigi 30

Vettori Piero 33

Zanetti Lorenzo 18

stampi:

Alberani Luigi

Vettori Piero

Zanetti Lorenzo

Voto medio: 27.0

Soluzione da dividere su più files

```
#include <stdio.h>

#define N 4

struct studente{

        char Nome[20];

        char Cognome[20];

        int Voto;};

void leggi(int n, struct studente Vet[]);

float media(int n, struct studente Vet[]);

main(){

    int i;

    struct studente V[N];

    leggi(N,V);

    for(i=0;i<N;i++)
    {

        printf("%s\n",V[i].Nome);

        printf("%s\n", V[i].Cognome);

    }

    printf("\nVoto medio: %f\n",media(N,V));

}

void leggi(int n, struct studente Vet[]){

    int i,j;
```


LIA

```
for(i=0;i<n;i++){  
    printf("Inserisci Nome, Cognome e Voto");  
    scanf("%s",Vet[i].Nome);  
    scanf("%s",Vet[i].Cognome);  
    scanf("%d",&Vet[i].Voto);  
}  
}  
  
float media(int n, struct studente Vet[]){  
    int i;  
    float m=0.0;  
    for(i=0;i<n;i++)  
        m=m+Vet[i].Voto;  
    return m/n;  
}
```