Funzioni 2

Esercitazione 8

30/10/2024 - Alessandro Montenegro

Funzioni

Interazioni con Array

```
void stampaArray(int v[], int dim){
   int i;
   for(i=0; i<dim ; i++){
      printf("%d", v[i]);
   }
}
```

Funzioni

Interazioni con Array

```
void stampaArray(int v[], int dim){
   int i;
   for(i=0; i<dim ; i++){
      printf("%d", v[i]);
   }
}
```

- Un array può essere passato in input
- Questo può avvenire senza specificare il numero di elementi
- Il compilatore interpreta int v[] come int * v
- Di fatto viene passato l'indirizzo della prima cella di memoria dell'array

Funzioni

Interazioni con Array

```
void stampaArray(int v[], int dim){
   int i;
   for(i=0; i<dim ; i++){
      printf("%d", v[i]);
   }
}
```

- La dimensione dell'array non è nota e quindi va passata come argomento!
- (A meno che non sia definita come costante)

Funzioni

Interazioni con Array

- Dato che si agisce su indirizza di memoria, si può modificare un array in una funzione
- Quando l’array è passato dal chiamante e viene modificato in una funzione, il chiamante vede le modifiche

Funzioni

Interazioni con Array

```
void stampaArray(int v[], int dim){
   int i;
   for(i=0; i<dim ; i++){
      printf("%d", v[i]);
   }
}
```

Prototipo

```
void stampaArray(int [], int);
```

Funzioni

Restituire Array?

?

?

?

?

?

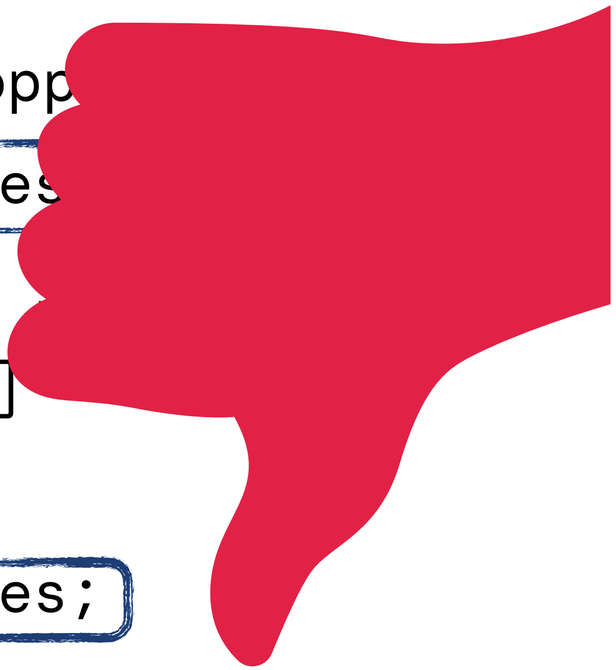
?

```
#define N 100
int * raddoppiaArray(int v[], int dim){
    int i, res[N];
    for(i=0; i<dim ; i++){
        res[i] = 2*v[i];
    }
    return res;
}
```

Funzioni

Restituire Array?

```
#define N 100
int * raddopp                    , int dim){
    int i, res
    for(i=0;
        res[i]
    }
    return res;
}
```



Funzioni

Restituire Array?

Tutto quello che è **locale alla funzione** viene **distrutto** quando la funzione termina

Gli array sono **aree di memoria che vengono de-allocate** quando viene invocato il “return”

Quindi il **chiamante** vedrebbe un’area di memoria con elementi a caso… un **dangling pointer**!

Funzioni

Restituire Array?

Come si fa?

Memoria dinamica

(Poi...)

Array statici

(Non li vediamo)

Definire un **array** nel **chiamante** per **ospitare** il **risultato** e passarlo come **argomento**

Funzioni

Restituire Array?

```
#define N 100
void raddoppiaArray(int v[], int dim, int res[]){
    int i;
    for(i=0; i<dim ; i++){
        res[i] = 2*v[i];
    }
    return;
}
```

Funzioni

Interazioni con Matrici

```
#define R …
#define C …
void fMatrice(int m[][C]){
   // …
}
```

Prototipo

```
#define R …
#define C …
void fMatrice(int [][C]);
```

Qui bisogna **esplicitare il numero di colonne** (ma si può omettere quello di righe)

Questo per come è **organizzato** un array multi-dimensionale in **memoria**

Il resto è come per gli array mono-dimensionali

Conta Occorrenze in un Array

Esercizio 1

Scrivere un programma che legge un array di interi e un numero e stampa il numero di volte che il numero appare all'interno dell'array.

Scrivere le funzioni:

- leggiInt(): che legge e restituisce un intero
- leggiIntN(): che legge e restituisce un intero minore di N e positivo
- leggiArrayInt(): che legge e restituisce un array di interi di una certa dimensione
- contaOccorrenze(): che prende in input un array e un numero e restituisce il numero di volte che il numero appare all'interno dell'array

Elimina Carattere da Stringa

Esercizio 2

Scrivere un programma che legge una stringa senza spazi e un carattere e stampa la stringa inserita senza il carattere inserito.

Scrivere le funzioni (da mappare con quanto possibile fare in C):

- leggiCarattere(): che legge e restituisce un carattere
- leggiStringa(): che legge e restituisce una stringa
- rimuoviCarattere(): che prende in input una stringa e un carattere e restituisce la stringa senza il carattere inserito

Lunghezza Massima Sottosequenza Ascendente

Esercizio 3

Scrivere un programma che prende in input nella funzione main una stringa. Il programma stampa la lunghezza della più lunga sottosequenza di caratteri ordinati in ordine ascendente nella stringa.

Scrivere la funzione:

- lunghezzaMassimaSottosequenzaAscendente(): che prende in input un puntatore che punta alla stessa area di memoria della stringa e la lunghezza della stringa. La funzione restituisce un intero pari alla più lunga sottosequenza di caratteri ordinati in ordine ascendente nella stringa.
- leggiStringa(): che legge e restituisce una stringa

Lunghezza Massima Sottosequenza Ascendente

Esercizio 3

Scrivere un programma che prende in input nella funzione main una stringa. Il programma stampa la lunghezza della più lunga sottosequenza di caratteri ordinati in ordine ascendente nella stringa.

Scrivere la funzione:

- lunghezzaMassimaSottosequenzaAscendente(): che prende in input un puntatore che punta alla stessa area di memoria della stringa e la lunghezza della stringa. La funzione restituisce un intero pari alla più lunga sottosequenza di caratteri ordinati in ordine ascendente nella stringa.
- leggiStringa(): che legge e restituisce una stringa

Elevare al Quadrato una Porzione dell'Array

Esercizio 4

Scrivere un programma che legge un array di lunghezza definita dall'utente (al massimo una dimensione predeterminata) e che modifica l'array elevando al quadrato una porzione di esso a partire da una posizione decisa dall'utente fino a quando l'array non termina o fino a quando non si incontra uno 0.

Scrivere le funzioni:

- leggiInt(): che legge e restituisce un intero
- leggiIntN(): che legge e restituisce un intero in [1,N]
- leggiArrayInt(): che legge e restituisce un array di interi di una certa dimensione
- foo(): che eleva al quadrato gli elementi dell'array da una certa posizione in poi fino ad arrivare alla fine dell'array o fino a quando l'array non termina. Tale funzione fa uso della funzione ausiliaria subpow()
- subpow(): che prende l'inizio di una porzione dell'array e la dimensione di tale sotto-array e lo eleva al quadrato
- stampaArray(): che stampa un array passato in input

Elevare al Quadrato una Porzione dell'Array

Esercizio 4

Scrivere un programma che legge un array di lunghezza definita dall'utente (al massimo una dimensione predeterminata) e che modifica l'array elevando al quadrato una porzione di esso a partire da una posizione decisa dall'utente fino a quando l'array non termina o fino a quando non si incontra uno 0.

Scrivere le funzioni:

- leggiInt(): che legge e restituisce un intero
- leggiIntN(): che legge e restituisce un intero in [1,N]
- leggiArrayInt(): che legge e restituisce un array di interi di una certa dimensione
- foo(): che eleva al quadrato gli elementi dell'array da una certa posizione in poi fino ad arrivare alla fine dell'array o fino a quando l'array non termina. Tale funzione fa uso della funzione ausiliaria subpow()
- subpow(): che prende l'inizio di una porzione dell'array e la dimensione di tale sotto-array e lo eleva al quadrato
- stampaArray(): che stampa un array passato in input

Elimina Sottostringa da Stringa

Esercizio 5

Scrivere un programma che prende in input una stringa e una seconda stringa. Il programma rimuove dalla prima stringa tutte le sequenze di caratteri uguali alla seconda stringa.

Scrivere le funzioni:

- leggiStringa()
- rimuoviStringa()

Shift di Matrice per Righe

Esercizio 6

Scrivere un programma che legge una matrice 3x4 ed effettua uno shift per righe verso il basso e stampa il risultato. L'ultima riga prende la posizione della prima riga.

Scrivere le funzioni:

- leggiArray()
- leggiMatrice()
- stampaArray()
- stampaMatrice()
- copiaArray()
- shiftMatrice()

Contatti

Alessandro Montenegro

Mail: alessandro.montenegro@polimi.it

Sito: https://montenegroalessandro.github.io/InfoA2425/index.html