

Esercizi 4IIN 24.09.2021 Ripasso

Classificazione dei **tipi di dato in C/C++**

- Tipi di dato
 - Scalari
 - predefiniti o base o primitivi: int, char, float, double, bool (C++)
 - enumerativi: enum
 - Derivati
 - strutturati: [] (array), struct (strutture o record), union
 - definiti dall'utente: typedef
 - puntatore
 - riferimento (in C++)

Regola pratica per stabilire la direzionalità di un parametro.

Porsi idealmente all'interno del sottopgm e chiedersi:

- Il valore del parametro viene fornito dall'esterno? (**IN**)
- Il valore del parametro viene fornito all'interno? (**OUT**)
- Il valore del parametro viene modificato all'interno del sottopgm? (**IN/OUT**)

Linguaggio C

- Passaggio di parametri per valore
- Passaggio di parametri per indirizzo

In C le variabili di tipi base (int, char, float, ...) si possono passare sia per valore che per indirizzo (dipende dalla direzionalità).

In C le variabili vettore, matrice e stringhe si possono passare solo per indirizzo.

In C le variabili di tipo record si possono passare sia per valore che per indirizzo. Noi le passeremo sempre per indirizzo per motivi di efficienza.

Linguaggio C++

- Passaggio di parametri per valore
- Passaggio di parametri by reference

In C++ le variabili di tipi base (int, char, float, ...) si possono passare sia per valore che by reference (dipende dalla direzionalità).

In C++ le variabili vettore, matrice e stringhe le passeremo sempre per indirizzo perché noi le abbiamo studiate in stile C.

In C++ le variabili di tipo record si possono passare sia per valore che by reference. Noi le passeremo sempre by reference per motivi di efficienza.

Linguaggio	Direzionalità	Tipo variabile	Modalità di passaggio
Linguaggio C	I	Tipi base	Per valore
		Vettori, matrici, stringhe	Per indirizzo (usa const)
		Record	Per indirizzo (usa const)
	O	Tipi base	Per indirizzo
		Vettori, matrici, stringhe	Per indirizzo
		Record	Per indirizzo
	I/O	Tipi base	Per indirizzo
		Vettori, matrici, stringhe	Per indirizzo
		Record	Per indirizzo
Linguaggio C++	I	Tipi base	Per valore
		Vettori, matrici, stringhe	Per indirizzo (usa const)
		Record	By reference (usa const)
	O	Tipi base	By reference
		Vettori, matrici, stringhe	Per indirizzo
		Record	By reference
	I/O	Tipi base	By reference
		Vettori, matrici, stringhe	Per indirizzo
		Record	By reference

Esercizio n.1 Vettori e sottopgm

Scrivi un sottopgm che dato un vettore di interi verifica se il vettore è ordinato in modo crescente.

Questo sottopgm è una funzione o una procedura?

Esercizio n.2 **Matrici e sottopgm**

Scrivi un sottopgm che data una matrice di interi ritorna il valore massimo e la sua posizione.

Questo sottopgm è una funzione o una procedura?

Esercizio n.3 **Stringhe e sottopgm**

Scrivi un sottopgm che data una stringa cancella tutti i caratteri alfabetici (sia maiuscoli che minuscoli) eseguendo un'operazione di shift verso sx.

Questo sottopgm è una funzione o una procedura?

Esercizio n.4 **Record e sottopgm**

- IL Body Mass Index (BMI) o Indice di Massa Corporea (IMC) è un parametro molto utilizzato per ottenere una valutazione generale del proprio peso corporeo.
- Esso mette in relazione con una semplice formula matematica l'altezza con il peso del soggetto.
- Si ottiene dividendo il peso in Kg del soggetto con il quadrato dell'altezza espressa in metri.
- Il risultato di tale formula classifica il soggetto in un'area di peso che può essere: normale - sottopeso - sovrappeso - obesità di medio grado - obesità di alto grado.

BMI	CONDIZIONE
< 16.5	GRAVE MAGREZZA
16-18,49	SOTTOPESO
18.5-24,99	NORMOPESO
25-29,99	SOVRAPPESO
30-34,99	OBESITÀ CLASSE I (lieve)
35-39,99	OBESITÀ CLASSE II (media)
≥ 40	OBESITÀ CLASSE III (grave)

Scrivi un sottopgm che riceve peso (in kg) e altezza (in cm) in una variabile di tipo record e ritorna il BMI.

Si legga una sequenza di coppie (peso, altezza) e per ciascuna coppia richiama il sottopgm che calcola e restituisce il BMI. La sequenza termina quando viene letta la coppia (0, 0).

Questo sottopgm è una funzione o una procedura?

Esercizio 5 Strutture dati – Vettore di record - Sottpgm

Si vuole realizzare un programma per la gestione degli ordini a un ristorante. Per ogni ordine bisogna memorizzare giorno e ora, numero di tavolo a cui si riferisce e elenco dei piatti richiesti, con eventuale quantità.

Ogni piatto è contraddistinto da un nome e da un costo, si supponga che ogni ordine non possa contenere più di 100 piatti diversi.

Per questo programma, dopo aver definito le opportune strutture in C/C++, si definiscano le seguenti funzioni:

- una funzione **stampaOrdine()** che, ricevuto come parametro un ordine, stampi a video tutti i dati che lo riguardano e anche il costo totale dei piatti che lo compongono. Questa funzione utilizza un'altra funzione che si chiama **totaleOrdine()**.
- una funzione **stampaOrdini()** che, ricevuti come parametri un vettore di ordini, la lunghezza del vettore, una data e un valore reale S, stampi a video tutti gli ordini che sono stati fatti in quella data e il cui costo totale sia superiore a S euro. Questa funzione utilizza tre funzioni **ordineInData()** che verifica se l'ordine corrente è nella data richiesta, **totaleOrdine()** e **stampaOrdine()**.

Prova a scrivere tu la struttura dati: dichiarazione tipi e costanti a livello globale.

Confronta la tua proposta con la mia che trovi nella pagina seguente.

Utilizza poi la mia proposta.

Scrivi i sottopgm richiesti e utilizza il mio **main()** per testarli.

I sottopgm sono funzioni o procedure?

```
#define MAX_PIATTI 100
```

```
#define DIM_STR 25+1
```

```
typedef struct {  
    char nomePiatto [DIM_STR];  
    float costo ;  
    int qta ;           // quantità  
} piatto ;
```

```
typedef struct {  
    int giorno , mese , anno ;  
    int ora , minuti ;  
} dataOra ;
```

```
typedef struct {  
    dataOra dataOrdine ;  
    int numeroTavolo ;  
    piatto piatti [ MAX_PIATTI ];  
    int piattiRichiesti ;  
} ordine ;
```

```

int main () {
    ordine ordini [] = {
        {
            {31 ,8 ,2020 ,14 ,30} ,
            5,
            {
                {" Penne " , 10, 2},
                {" Tortellini " , 12, 3},
                {" Tagliata " , 18, 1}
            }
        },
        3
    },
    {
        {31 ,8 ,2020 ,16 ,30} ,
        7,
        {
            {" Penne " , 10, 1},
            {" Spaghetti " , 8, 1},
            {" Stufato " , 22, 2},
            {" Torta " , 4, 2}
        }
    },
    4
},
{
    {30 ,8 ,2020 ,16 ,30} ,
    5,
    {
        {" Cannelloni " , 10, 7},
        {" Torta " , 8, 7}
    }
},
2
}
};

```

```

    int numOrdini = sizeof ( ordini ) / sizeof ( ordine );

    printf ( " Stampa del primo ordine \n");
    ...

    printf ( " Stampa ordini in data 31/08/2020 \n");
    dataOra d = {31 ,8 ,2020 ,16 ,30};
    ...

    return 0;
}

```

Esercizio 6 **Stringhe - Sottopgm**

Scrivi un sottopgm che data una stringa s contenente anche delle cifre, sommi tutte le cifre e fornisca il valore totale di tale somma.

Esempio:

s="abc3x casa2 y34zq" il programma deve restituire 3+2+3+4=12.

Il sottopgm è una funzione o una procedura?

Esercizio 7 **Vettore di stringhe**

Scrivi un sottopgm che ordina in modo crescente un vettore di stringhe. Si utilizzi l'algoritmo studiato lo scorso anno: "Selection sort".

Nella letteratura informatica esistono molti algoritmi di ordinamento. Cerca in internet l'algoritmo "Bubble sort" e applicalo nell'esercizio di prima ordinando l'array per lunghezza decrescente delle stringhe.

Il sottopgm è una funzione o una procedura?

Esercizio 8 **Ricorsione**

Scrivi una funzione ricorsiva che calcola la somma degli elementi di un array di interi di numElem elementi.

Esercizio 9 **Ricorsione**

Scrivere una funzione ricorsiva che calcola, dati due numeri interi M (>0) ed N (≥ 0), la potenza M^N .

Esercizio n.10 **File di testo**

Si desidera sviluppare un programma in linguaggio C++ per il calcolo delle superfici e dei volumi di un edificio.

Il programma riceve in input il nome del file che contiene le dimensioni dell'edificio.

La struttura dell'edificio è descritta in un file di testo così organizzato.

La prima riga del file contiene il numero dei piani dell'edificio e per ogni piano è presente una riga che contiene due valori interi: il numero di stanze presenti nel piano e l'altezza del piano. Tale riga è seguita da tante righe quante sono le stanze, ognuna contenente due valori che rappresentano le dimensioni della stanza.

Tutte le stanze sono di forma rettangolare, tutte le dimensioni sono espresse in centimetri e sono date come numeri interi positivi.

Il programma deve calcolare e presentare sull'unità di output standard:

- la superficie totale di tutte le stanze dell'edificio, espressa in metri quadri
- il volume totale di tutte le stanze dell'edificio, espresso in metri cubi

Esempio:

Input: edificio.txt

2

2 300
200 200
200 400
1 200
200 300

(ovvero il primo piano è alto 300 cm e consiste di due stanze rispettivamente di 200 cm × 200 cm e 200 cm × 400 cm, mentre il secondo piano è alto 200 cm e consiste di un'unica stanza di 200 cm × 300 cm)

Output:

Superficie totale dell'edificio: 18.00 metri quadri

Volume totale dell'edificio: 48.00 metri cubi