

Esercitazione 8

Nella propria home directory creare una sottodirectory chiamata es08, in cui metteremo tutti i file C di oggi.

Esercizi su procedure ricorsive

1) Scrivere una funzione ricorsiva che riceva un intero e stampi ricorsivamente un numero corrispondente di punti '.'

2) Scrivere una funzione ricorsiva che riceva un array di interi e ritorni il valore calcolato sommando tutti gli elementi pari e sottraendo tutti gli elementi dispari.

3) Scrivere una funzione ricorsiva che riceva un array di caratteri e un carattere e ritorni la posizione del primo carattere non minore di quello passato. Se non c'è, la funzione ritorni -1.

4) Scrivere una funzione ricorsiva che calcoli il coefficiente binomiale (n,m) sfruttando le seguenti relazioni (derivate dal triangolo di Tartaglia):

$$(n,m) = (n-1, m-1) + (n-1, m)$$

$$(n,n) = (n,1) = 1$$

con $1 \leq n$ e $1 \leq m \leq n$

5) Scrivere una funzione ricorsiva che ricevuto un array di caratteri, restituisca il carattere più piccolo in ordine alfabetico.

6) Scrivere una funzione ricorsiva che, dato un array di interi, inverta le posizioni dei suoi elementi.
Ad esempio l'invocazione `inverti(a)`, dove `a` è l'array `{1, 2, 3, 4}`, deve modificare l'array in `{4, 3, 2, 1}`.

7) Sulla base dell'esercizio 1, riuscite a far stampare ricorsivamente punti '.' alternati a '_'? E punti '.' alternati a '_' e a '='?

Qual'è il metodo generale?

Esempio di interazione col programma

Inserire un numero: 7

7 -> ._._.

Inserire un numero: 9

9 -> ._.=._.=.

8) Scrivere una funzione iterativa e ricorsiva che presi in input due vettori A e B lunghi n stampi insieme il primo valore di A e l'ultimo valore di B, poi il secondo valore di A e il penultimo di B e così

via.

9) Scrivere una funzione merge che presi in input due array ordinati li fonda in un unico array ordinato. Tale funzione avrà tra i parametri tre array, due sono quelli da fondere ed il terzo conterrà il risultato. Implementare la versione iterativa e ricorsiva.

10) Dati due vettori di interi A e B di lunghezza n, si ottenga di scambiare il primo elemento del vettore A con l'ultimo del vettore B, il secondo di A con il penultimo di B e così via. Per scambiare gli elementi si faccia uso di una funzione a cui siano passati solo gli elementi da scambiare e non l'intero vettore.

Definire una funzione **scambia**.

Richiamare la funzione scambia sugli opportuni valori dei vettori

Esercizi strutture:

Attraverso le strutture potete mettere insieme più cose di tipo diverso. Esempio:

```
struct anagrafe {
    char[100] nome;
    char[100] cognome;
    int eta;
    int numeroFigli;
    char sesso;
};
```

Come dichiaro un elemento di questa struttura, allo stesso modo dei tipi primitivi.

Per int avevamo: `int x`, oppure `int x[N]` per i vettori

Ora abbiamo:

```
struct anagrafe x, struct anagrafe x[N]
```

Le uniche operazioni valide sulle strutture sono:

- Assegnare una struttura ad un'altra struttura dello stesso tipo
- rilevare l'indirizzo (&) di una variabile di tipo struttura
- accedere ai membri di una variabile di tipo struttura
- usare l'operatore sizeof per determinare la dimensione di una struttura

N.B. le strutture non si possono confrontare con `==` e `!=` (Vanno confrontati i singoli elementi che compongono la struttura)

Le strutture possono essere passate alle funzioni:

- Passando l'intera struttura
- Passando singoli membri

In entrambi i modi il passaggio è per valore

Per passare le strutture per riferimento, passare l'indirizzo.

Un vettore di strutture è passato per riferimento (è un vettore!) creando una struttura contenente un vettore e passandola ad una funzione si ottiene il passaggio di un vettore per valore (se non si passa il riferimento della struttura e si modifica il vettore all'uscita della funzione il vettore della struttura rimane lo stesso)

11) Definire un nuovo tipo di dato capace di rappresentare una data. Scrivere poi delle opportune funzioni/procedure che:

- ricevuta una data la aggiorni al giorno successivo (ignorando gli anni bisestili);
- ricevute due date verifichino che la prima preceda la seconda;
- ricevute due date, ritornino la differenza in anni fra la prima e la seconda.

12) Definire un nuovo tipo di dato capace di rappresentare i dipendenti di una ditta: di tali dipendenti interessa il codice numerico identificativo unico, la data di assunzione, il grado raggiunto (un intero) e lo stipendio.

Usare il tipo di dato definito nell'esercizio precedente per tutte le date.

Scrivere le seguenti funzioni/procedure:

- `calcolaAnzianita` che, dato un dipendente, calcoli il numero di anni trascorsi dalla data di assunzione al 2102;
- `aggiornaStipendio` che, dato un dipendente, aumenti del 1% il suo stipendio per ogni anno di anzianita' accumulato (attenzione l'interesse da calcolare e' un interesse composto).
- `superiore`, che dati due dipendenti ritorni 1,0 o -1 se il primo e' rispettivamente un superiore, un pari grado o un sottoposto del secondo.

Notare che a parita' di grado, il superiore e' il dipendente con l'anzianita' maggiore. Se e' pari anche l'anzianita', i dipendenti sono pari grado.

Scrivere quindi un programma che richieda all'utente di inserire i dati di

due impiegati, utilizzi le due procedure di cui sopra e ne stampi l'esito.