

Esercitazioni di Informatica B

Struct

Stefano Cereda

stefano.cereda@polimi.it

30/10/2018

Politecnico di Milano



Un supermercato ha memorizzato il proprio archivio di scontrini nell'array *archivio*, in cui ogni elemento è una struttura dati di tipo *scontrino* con le seguenti informazioni:

- IDcliente: stringa di 10 caratteri che identifica univocamente il cliente
- totale: totale della spesa in Euro
- punti: punti premio associati alla spesa

Ad ogni spesa, vengono assegnati ad ogni cliente un quantitativo di punti premio pari alla somma dei punti raccolti più un ulteriore punto premio per ogni 10 euro spesi.

Scontrini ii

1. Definire il tipo *scontrino* e l'array *archivio*
2. Scrivere il frammento di codice C che, dato il codice identificativo di un cliente, calcoli la somma dei suoi punti premio.
3. Scrivere il frammento di codice C che ordina l'array *archivio* secondo il totale della spesa in ordine decrescente.

Nota: Si faccia attenzione al fatto che un cliente può comparire in più di uno scontrino.

Si vuole gestire l'anagrafe della motorizzazione e dei proprietari di auto. Un veicolo è identificato da:

- Targa
- Modello
- Proprietario (si suppone che ogni auto abbia un unico proprietario)

I proprietari di auto sono identificati dai campi:

- Cognome
- Nome
- Un insieme di indirizzi di recapito (max 3).

Un recapito è composto da: via, numero civico e CAP. L'anagrafe contiene al max N veicoli.

Anagrafe

Definire in C la struttura dati che permette di rappresentare la situazione descritta.

Scrivere il programma C che chiede all'utente di inserire i dati degli N veicoli da memorizzare nell'anagrafe. Per ogni veicolo il programma deve acquisire anche le informazioni relative al proprietario, inclusi i suoi recapiti. In una prima versione del programma si inseriscano esattamente tre recapiti per proprietario, in una seconda versione si consenta all'utente di inserire un numero di recapiti minore o uguale a tre.

Finita la fase di inserimento il programma stampa il contenuto della struttura nel formato:

```
VEICOLO Targa XXXXX  
PROPRIETARIO Cognome XXXXXXXXX  
RECAPITI CAP XXXXXXXXX ... XXXXXXXXX
```

Definire un tipo di dato *fantacalcio* che contiene una lista di $N = 20$ *fantasquadre*. Ogni *fantasquadra* è identificata dal nome, il colore della casacca, il punteggio corrente e l'allenatore. L'*allenatore* è definito dal nome, cognome e numero di coppe vinte (un intero ≥ 0). Infine si dichiara una variabile *campionato* di tipo *fantacalcio*.

Scrivere un frammento di codice in linguaggio C che stampi a video il numero totale degli allenatori (contenuti all'interno della variabile *campionato*) che hanno vinto almeno una coppa.

Un'organizzazione che si occupa di archeologia sottomarina vuole sviluppare un software in linguaggio C che permetta di catalogare tutti i reperti archeologici che sono stati scoperti. Ogni reperto è caratterizzato dalle seguenti informazioni:

- coordinate geografiche in cui si trova
- profondità del reperto rispetto al livello del mare
- peso del reperto
- tipo di reperto, questo può essere: manufatto in terracotta, manufatto in marmo, manufatto in metallo

Tutti i reperti vengono memorizzati in un array di dimensione massima 1000.

1. Si definiscano in linguaggio C le strutture dati e le variabili necessarie per rappresentare l'insieme dei reperti nella variabile *insiemeReperti*.
2. Si supponga che la variabile *insiemeReperti* sia stata precedentemente riempita con le informazioni di 1000 reperti e si sviluppi in C il ciclo che, per ogni reperto, stampa a video il suo peso. Si dichiarino anche tutte le variabili necessarie allo scopo che non sono state dichiarate nel punto precedente.

Periodicamente, l'organizzazione progetta spedizioni per raccogliere i reperti scoperti. L'operazione non è sempre possibile perché il veicolo che viene usato per raccogliere i reperti ha una portata massima (un peso massimo che

può portare) che non può essere superato. D'altra parte, l'organizzazione vuole fare in modo che il veicolo venga utilizzato nel modo più efficiente possibile e quindi la spedizione verrà pianificata in modo tale che il veicolo raccolga il maggior numero possibile di reperti nell'area prescelta. La spedizione viene pianificata come segue: Si parte con un veicolo scarico, di portata p_{Max} , da una coppia di coordinate geografiche (*latitudinel* e *longitudinel*) e si esplorano le vicinanze di quelle coordinate geografiche in modo da rimanere nell'ambito della zona di ricerca, i cui confini sono definiti dai punti *latitudinel*soglia, *longitudinel*soglia. Per ogni reperto che si trova nei confini definiti, si pianifica di caricare il

reperito a bordo del veicolo, a patto che il suo peso, sommato al peso del carico corrente portato dal veicolo, sia minore di $pMax$. In caso contrario, quel reperto viene scartato e si procede a cercarne un altro, sempre rimanendo nei confini dell'area di ricerca. La ricerca termina quando il peso dei reperti da trasportare avrà raggiunto il valore $pMax$ oppure quando tutti i reperti nella zona di ricerca saranno stati esaminati.

L'algoritmo di pianificazione del viaggio, durante l'analisi sopra descritta, crea un array di reperti chiamato *caricoVeicolo* che contiene, alla fine dell'analisi, tutti i reperti che dovranno essere caricati sul veicolo durante l'esecuzione della spedizione.

3. Dato un reperto che assumiamo si trovi nella cella i -esima (con i che ha un valore qualsiasi compreso tra 0 e 999, estremi inclusi) dell'array *insiemeReperti*, si scriva la condizione che, se vera, ci permette di dire che il reperto si trova nell'area di ricerca definita dai confini indicati sopra.
4. Si sviluppi il frammento di codice in C necessario per acquisire da tastiera: la portata massima del veicolo, le coordinate geografiche di partenza della spedizione, il valore soglia che definisce i confini di ricerca. Si dichiarino tutte le variabili necessarie allo scopo.

5. Si sviluppi in linguaggio C l'algoritmo di pianificazione del carico descritto in precedenza. Oltre alle dovute inizializzazioni di variabili (non è necessario inizializzare *insiemeReperti* che si assume essere già riempito), la parte principale di questo algoritmo sarà costituita a un ciclo che analizza tutti i reperti in *insiemeReperti* e, per ogni reperto:

- Valuta se si trova nell'area di ricerca (si faccia riferimento alla soluzione fornita per il punto 3)
- In caso positivo, valuta se può essere caricato sul veicolo
- In caso positivo, aggiorna la variabile *caricoVeicolo* in modo che contenga i dati del reperto in questione

Si dichiarino anche tutte le variabili necessarie allo scopo che non sono state dichiarate nei punti precedenti.