

Esercizi 3AIIN 07.05.2021

- Salva in Moodle un file zip che contiene solo i pgm sorgenti.
- Formato nome file: **CognomeNomeGGMMAAAA.estensione**

dove GGMMAAAA è la data di assegnazione del lavoro.

- Terminata ogni lezione di laboratorio devi caricare in Moodle il lavoro svolto.
- A casa eventualmente puoi finirlo e/o correggerlo. Hai tempo una settimana.
- Anche chi è assente il giorno della consegna è tenuto a svolgere il lavoro assegnato.
- Verranno fatti controlli a campione e sicuramente durante le interrogazioni.

Esercizio n. 1

Scrivere un pgm C/C++ che gestisce un array di NMAX elementi; ciascun elemento è costituito da una stringa **s** di DIM_STR caratteri e da un intero **c**.

L'array è inizialmente vuoto e la gestione avviene accettando da input una serie di comandi. I comandi sono:

I/i - Inserimento (ordinato): accettazione da input di una stringa **ns**; se non esiste alcun elemento con **s = ns** si inserisce un nuovo elemento nell'array in cui si memorizza **ns** in **s** e si pone **c** ad uno (la stringa deve essere inserita in modo da mantenere la struttura dati ordinata in modo crescente rispetto il campo che contiene la stringa, senza chiamare un sottopgm di ordinamento); in caso contrario si incrementa **c** di una unità nell'elemento **ns = s**.

C/c - Cancellazione: accettazione da input di una stringa **ns**; se esiste un elemento con **s = ns** si decrementa **c** di una unità; se **c** raggiunge il valore zero si elimina l'elemento dall'array attraverso un'operazione di shift degli elementi verso sinistra; se l'elemento non esiste si visualizza un messaggio d'errore.

V/v - Visualizzazione di tutti di elementi presenti nell'array in ordine decrescente di occorrenze. La struttura dati (principale) non deve essere modificata. Si richiede quindi di copiare la struttura dati in un'altra e su questa effettuare l'ordinamento come richiesto.

Layout stampa:

stringaX occX

... ...

R/r - Data una stringa stampare il rispettivo numero di occorrenze. Devi applicare la ricerca binaria o dicotomica e ritornare l'indice della componente della struttura dati.

F/f - Fine lavoro.

Rilevare le eventuali condizioni di errore e visualizzare (nel main()) un messaggio opportuno.

Sottogm richiesti (con parametri):

- menu(): presenta sul video il menu dei comandi permessi.
- inserimento(): inserisce una stringa nella struttura dati (v. algoritmo sopra). Utilizza cercaStr().

- `cancellazione()`: cancella una stringa dalla struttura dati (v. algoritmo sopra). Utilizza `cercaStr()`.
- `cercaStr()`: ricerca (in modo sequenziale) nella struttura dati ordinata la stringa da inserire o da cancellare. La funzione di ricerca ritorna la posizione (0, 1, 2, ...) o -1 se non trovata. Codifica la ricerca tenendo conto che la struttura dati è ordinata.
- `visualizzazione()`: visualizza il contenuto della struttura dati (v. modalità sopra). Utilizza `copia()` e `sort()`.
- `copia()`: copia tutti i dati contenuti nella struttura dati (principale) in una variabile locale al sottopgm `visualizzazione()`.
- `sort()`: ordina tutti gli elementi presenti nella struttura dati (copia) in ordine decrescente di occorrenze.
- `ricercaBinaria()`: applica l'algoritmo di ricerca binaria. Ritorna l'indice della componente della struttura dati, -1 se non trovato. Scrivi sia la versione iterativa che ricorsiva.

NOTA BENE:

- Per il test del programma può esserti utile un sottopgm che esegue la stampa della struttura dati (`stampa()`).
- I sottopgm, eccetto `visualizzazione()` ed eventualmente `stampa()`, **NON** devono eseguire operazioni di I/O.
- Non effettuare controlli sull'input delle stringhe lette. Si suppone che $\text{strlen}(s) \leq 25$.

<pre>#define DIM_STR 25+1 #define NMAX 50 typedef struct{ char s[DIM_STR]; int c; }elem; int main(){ elem lista[NMAX]; int numElem=0; }</pre>	<pre>int menu(){ ... cout<<"\n\n G E S T I O N E S T R I N G H E \n\n"; cout<<"I/i - Inserimento \n"; cout<<"C/c - Cancellazione \n"; cout<<"V/v - Visualizzazione \n"; cout<<"R/r - Ricerca \n"; cout<<"F/f - Fine lavoro \n\n"; cout<<"Digita una scelta \n"; ... return scelta; }</pre>
---	---

Pulizia del buffer: `while(cin.get()!='\n');`

Esempio:

`NMAX = 10`

All'inizio `numElem = 0`, dopo alcuni inserimenti questo è lo stato della struttura dati (vettore di record).

isola	mare	palma	sole	vacanza					
3	4	1	5	12					
0	1	2	3	4	5	6	7	8	9

numElem = 5