

Esame di Programmazione

Ing. Cristiano Gregnanin

July 6, 2015

1 Esercizio

Un programma deve organizzare le agende di un mese di un centro screening. Il centro screening ha i dati dei pazienti da invitare (al massimo 30), memorizzati nel file di testo *pazienti.txt*, che contiene i seguenti dati:

- cognome: stringa di 20 max caratteri
- priorità (intero); questo campo rappresenta la priorità con cui deve essere chiamato il paziente e può assumere valore 1 (paziente ad alto rischio), valore 2 (paziente a rischio normale), valore 3 (paziente a basso rischio) oppure valore 4 (paziente a rischio nullo)
- giorno dell'ultima visita (intero da 1 a 30)

Esempio di file *pazienti.txt*:

greco	2	5
bavaresco	2	7
tenan	3	9
stocco	1	12
piovan	3	20
zambello	4	21
schiesaro	1	22
merlo	4	24
gregnanin	1	25

Il file *pazienti.txt* è ordinato in ordine crescente per il campo giorno. Il programma deve scrivere su un file di testo *inviti.txt* gli appuntamenti che devono essere dati ai pazienti. Gli appuntamenti hanno la seguente struttura:

- cognome (stringa di 20 caratteri)
- priorità
- data (intero da 1 a 30).

Sapendo che ogni giorno del mese può essere visitato un paziente, si compilino le agende fissando prima le visite dei pazienti ad alto rischio, poi quelli a rischio normale, poi quelli a basso rischio e infine quelli a rischio nullo, secondo il seguente algoritmo:

1. si scriva una funzione *leggi* che riceve in input il nome del file da leggere e ritorni un'opportuna lista *prenotazioni* che ne rappresenti il contenuto.
2. Si scriva una funzione *raggruppaPerPriorita* che prende in ingresso un intero *priorita* che rappresenta una classe di rischio (da 1 a 4) e la lista *prenotazioni* e che fornisca in uscita una lista di cognomi dei pazienti che hanno come priorità il valore *priorita* dato in input. Ad esempio, se *priorita*=1, la funzione deve fornire nella lista di uscita tutti i pazienti che hanno priorità 1.
3. si scriva una funzione *bubblesort* che data la lista *prenotazioni* la ordini per cognome. Si utilizzi l'algoritmo bubblesort
4. Si scriva una funzione *organizzaAppuntamenti* che prende in ingresso la lista *prenotazioni* e che scriva sul file di testo *inviti.txt* i pazienti in ordine di priorità. Per fare questo, per ciascun valore di priorità (da 1 a 4) si invochi la funzione *raggruppaPerPriorita* e si salvi sul file *inviti.txt* il cognome del paziente, la sua priorità e la data dell'appuntamento. La data dell'appuntamento deve essere un numero progressivo (il primo paziente viene chiamato il 1 del mese, il secondo il 2, ecc.).

Esempio di file *inviti.txt*:

gregnanin	1	1
schiesaro	1	2
stocco	1	3
bavaresco	2	4
greco	2	5
piovan	3	6
tenan	3	7
merlo	4	8
zambello	4	9

5. si scriva una funzione *stampa* che prende in ingresso la lista *prenotazioni* e la stampi in modo opportunamente formattato stampando prima il cognome poi la priorità e per ultima la data.
6. si scriva una funzione *ricercaGiornoLibero* che prende in ingresso un intero *giorno* e la lista *prenotazioni* e ricerchi nella lista *prenotazioni* il *giorno*. Si utilizzi l'algoritmo di ricerca binaria per effettuare la ricerca. La funzione deve ritornare -1 se il giorno non è stato trovato
7. nel programma principale si richiami la funzione *leggi* e si salvi il suo output nella variabile *prenotazioni*
8. nel programma principale si richiami la funzione *bubblesort* e se ne calcoli il tempo di esecuzione stampandolo a video. (ad esempio: "il tempo di esecuzione di bubblesort è: 3.933452345e - 05")
9. nel programma principale si richiami la funzione *stampa* al fine di stampare la lista *prenotazioni*
10. nel programma principale si richiami la funzione *organizzaAppuntamenti*

11. nel programma principale si chieda in input all'utente un intero e lo si memorizzi nella variabile *data*
12. nel programma principale si richiami la funzione *leggi* dandole in input il file *inviti.txt* generato precedentemente e si salvi il suo output nella variabile *prenotazioni*
13. nel programma principale si richiami la funzione *ricercaGiornoLibero* dandole in input la variabile *data* e la lista *prenotazioni* e si salvi il suo output nella variabile *libero*. A questo punto se *libero* è maggiore uguale di 0 (ovvero se la ricerca è andata a buon fine) si stampi a video la stringa *il giorno è occupato* altrimenti si stampi la stringa *il giorno è libero*

2 Osservazioni

- In tutte le funzioni è possibile aggiungere parametri.
- Usare la funzione `split()` per estrarre il cognome, la priorita e la data. Si testi in console il suo funzionamento. Ad esempio si testi in console: `"merlo 4 8".split()`
- La creazione dei file di testo è a cura dello studente.
- Lo studente, se lo ritiene necessario, può implementare e utilizzare altre funzioni oltre a quelle previste dal testo.
- Per eseguire la misurazione si può utilizzare la funzione `clock()` della libreria *time*
- Convertendo un double in un integer si ottiene la sua parte intera. Ad esempio si testi in console: `int(1.5)`