Informatica – I Facoltà di Ingegneria

Prova scritta del 13/02/2014

Si chiede di realizzare un programma per la gestione del palinsesto settimanale di una emittente radiofonica. I dati del palinsesto settimanale sono mantenuti in un file denominato PALINSESTO.TXT in cui ogni riga rappresenta le informazioni di un evento programmato per la settimana nel seguente formato:

<giorno_della_settimana> <ora_inizio> <ora_fine> <titolo>

dove:

- Il campo <giorno_della_settimana> indica il giorno della settimana dell'evento ed è un numero intero da 1 a 7 (con 1 per il lunedì e 7 per la domenica)
- Il campo <ora_inizio> indica l'ora del giorno in cui inizia l'evento ed è un numero intero da 0 a 24 (con 0 per la mezzanotte)
- Il campo <ora_fine> indica l'ora del giorno in cui finisce l'evento ed è un numero intero da 0 a 24
- Il campo <titolo> è una stringa di caratteri (al massimo 80) senza spazi all'interno
- Il numero massimo di righe è 168 (7x24 essendo un'ora la durata minima di un evento)
- Le righe del file non seguono nessun ordinamento

Il programma all'avvio carica dal file le informazioni sul palinsesto e, dopo aver eseguito le operazioni indicate dall'utente, deve salvare il file aggiornato.

Il programma, tramite menu permette all'utente le seguenti operazioni:

1. Lista degli eventi della settimana, uno per riga, ordinati per giorno e ora di inizio (nello stesso formato indicato precedentemente <giorno_della_settimana> <ora_inizio> <ora_fine> <titolo>)
2. Inserimento di un nuovo evento
3. Uscita

Nel caso dell'operazione n.2 il programma:

- chiederà all'utente di inserire i dati dell'evento (sempre nel formato <giorno_della_settimana> <ora_inizio> <ora_fine> <titolo>)
- verificherà che l'evento non abbia sovrapposizioni con altri eventi (se ha sovrapposizioni comunicherà il messaggio "Errore, evento in sovrapposizione.")
  N.B. se un evento finisce alle 13 e un altro inizia alle 13 (cioè alla stessa ora) non sono da intendersi in sovrapposizione (perché l'estremo <ora_fine> e' da intendersi escluso).
- aggiungerà il nuovo evento in fondo al file in una nuova riga e stamperà il messaggio "Evento inserito."

Esempio:

Esecuzione del programma	File PALINSESTO.TXT (inizio)
1. Lista eventi 2. Inserimento evento 3. Uscita > 1 2 8 9 Giornale_radio 2 12 14 Il_pranzo_e_servito 3 8 9 Giornale_radio 7 12 13 Angelus 7 13 17 Tutto_il_calcio_minuto_per_minuto > 2 2 9 12 Mattina_in_famiglia Evento inserito. > 2 7 12 14 Il_pranzo_e_servito Errore, evento in sovrapposizione. > 3	2 8 9 Giornale_radio 7 13 17 Tutto_il_calcio_minuto_per_minuto 3 8 9 Giornale_radio 2 12 14 Il_pranzo_e_servito 7 12 13 Angelus File PALINSESTO.TXT (fine) 2 8 9 Giornale_radio 7 13 17 Tutto_il_calcio_minuto_per_minuto 3 8 9 Giornale_radio 2 12 14 Il_pranzo_e_servito 7 12 13 Angelus 2 9 12 Mattina_in_famiglia

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define FILENAME "palinsesto.txt"

#define TRUE    1
#define FALSE   0

#define NMAX    (24*7)
#define STRMAX  80

#define NOTUSED -1

struct palinsesto_t {
        int idx;
        int day;
        int start;
        int end;
        char title[STRMAX+1];
} palinsesto [NMAX];

/*
   Insert full information in the first slot (day:start)
   Unset NOTUSED in the following slots till day:end
*/
void insert(int day, int start, int end, char *title) {
        int idx_start, idx;

        // Get start index for the event
        idx_start = ((day-1)*24+start-1);

        // Save event data
        palinsesto[idx_start].idx = idx_start;
        palinsesto[idx_start].day = day;
        palinsesto[idx_start].start = start;
        palinsesto[idx_start].end = end;
        strcpy(palinsesto[idx_start].title,title);

        // set next records as used
        for(idx=idx_start+1;idx<((day-1)*24+end-1);idx++) {
                palinsesto[idx].idx = idx_start;
        }
}


int main(void) {

        FILE *fp;

        int scelta, busy;

        int i;
        int day, start, end;
        char title[80];

        if( (fp=fopen(FILENAME,"a+")) == NULL ) {
                printf("File %s not found.\n", FILENAME);
                exit(EXIT_FAILURE);
        }

        /* Set all records to not used */
        for(i=0;i<NMAX;i++) { palinsesto[i].idx = NOTUSED; }

        while( fscanf(fp, "%d %d %d %s", &day, &start, &end, title) != EOF ) {
                insert(day, start, end, title);
        }


        do {
                printf("1. Lista eventi\n");
                printf("2. Inserisci evento\n");
                printf("0. Esci\n");
                scanf("%d", &scelta);
```

```
                switch(scelta) {
                        case 1:
                                for(i=0; i<NMAX; i++) {
                                        // if idx is not NOTUSED and there is a title, then print it
                                        if(palinsesto[i].idx != NOTUSED && strlen(palinsesto[i].title)!
=0) {
                                                printf("%4d %4d %4d   %s\n", palinsesto[i].day,
palinsesto[i].start, palinsesto[i].end, palinsesto[i].title);
                                        }
                                }
                                break;
                        case 2:
                                printf("Inserisci nuovo evento: <giorno_settimana> <ora_inizio>
<ora_fine> <titolo>\n");
                                scanf("%d %d %d %s", &day, &start, &end, title) ;

                                /* check if the records that belongs to this event are busy or not */
                                busy = FALSE;
                                for(i=(((day-1)*24)+start-1); i< (((day-1)*24)+end-1); i++) {
                                        if(palinsesto[i].idx != NOTUSED) {
                                                busy = TRUE;
                                                break;
                                        }
                                }

                                // if all records are not busy insert the event
                                if(!busy) {
                                        insert(day, start, end, title);
                                        if( fprintf(fp, "%d %d %d  %s", day, start, end, title) > 0 ) {
                                                printf("Evento inserito.\n");
                                        }
                                } else {
                                        printf("Errore, evento in sovrapposizione.\n");
                                }

                                break;
                        default:
                                break;

                }

        } while(scelta == 1 || scelta == 2);

        fclose(fp);

        return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define FILENAME "palinsesto.txt"

#define TRUE    1
#define FALSE   0

#define NMAX    (24*7)
#define STRMAX  80

#define NOTUSED -1

struct palinsesto_t {
        int id;
        int day;
        int start;
        int end;
        char title[STRMAX+1];
} palinsesto [NMAX];

/*
   0: free slot
   id: slot busy with event id-1
*/
int timetable[7][24];
int id = 0;

/*
   Inserts the event in the palinsesto array
   Mark the timetable (from start to end hour) with the id of the event
*/
void insert(int day, int start, int end, char *title) {
        int i;

        /* Use id as index of the event (starts at 1) */
        id = id + 1;

        /* Save event data */
        palinsesto[id-1].id = id;
        palinsesto[id-1].day = day;
        palinsesto[id-1].start = start;
        palinsesto[id-1].end = end;
        strcpy(palinsesto[id-1].title,title);

        /* Mark timetable with the id of the event */
        timetable[day-1][start-1] = id;
/*      printf("Insert %s at %d:%d-%d.\n", title, day, start, end); */

        /* Mark next timetable slots with the id of the event * -1 */
        for(i=start+1;i<end;i++) {
                timetable[day-1][i-1] = id * -1;
        }

}


int main(void) {

        FILE *fp;

        int scelta, busy;

        int i,j,k;
        int day, start, end;
        char title[80];

        // reset timetable
        for(i=0;i<7;i++)
                for(j=0;j<24;j++)
                        timetable[i][j] = 0;

        if( (fp=fopen(FILENAME,"a+")) == NULL ) {
```

```
                printf("File %s not found.\n", FILENAME);
                exit(EXIT_FAILURE);
        }

        while( fscanf(fp, "%d %d %d %s", &day, &start, &end, title) != EOF ) {
                insert(day, start, end, title);
        }


        do {
                printf("1. Lista eventi\n");
                printf("2. Inserisci evento\n");
                printf("0. Esci\n");
                scanf("%d", &scelta);

                switch(scelta) {
                        case 1:
                                /* Scan the timetable and print the events */
                                for(i=0;i<7;i++) {
                                        for(j=0;j<24;j++) {
                                                if( (k=timetable[i][j]) > 0) {
                                                        /* if there is an id in that hour slot, print
the event */
                                                        /* print the event title only when id > 0
(avoid printing the same event multiple times, one for each hour slot) */
                                                        printf("%4d %4d %4d   %s\n", palinsesto
[k-1].day, palinsesto[k-1].start, palinsesto[k-1].end, palinsesto[k-1].title);
                                                }
                                        }
                                }
                                break;
                        case 2:
                                printf("Inserisci nuovo evento: <giorno_settimana> <ora_inizio>
<ora_fine> <titolo>\n");
                                scanf("%d %d %d %s", &day, &start, &end, title) ;

                                /* check if the records that belongs to this event are busy or not */
                                busy = FALSE;
                                for(j=start;j<end;j++) {
                                        if(timetable[day-1][j-1] != 0) {
                                                busy = TRUE;
                                                break;
                                        }
                                }

                                // if all records are not busy insert the event
                                if(!busy) {
                                        insert(day, start, end, title);
                                        if( fprintf(fp, "%d %d %d  %s", day, start, end, title) > 0 ) {
                                                printf("Evento inserito.\n");
                                        }
                                } else {
                                        printf("Errore, evento in sovrapposizione.\n");
                                }

                                break;
                        default:
                                break;

                }

        } while(scelta == 1 || scelta == 2);

        fclose(fp);

        return EXIT_SUCCESS;
}
```

```
2  8  9  Giornale_radio
7 13 17  Tutto_il_calcio_minuto_per_minuto
3  8  9  Giornale_radio
2 12 14  Il_pranzo_e_servito
7 12 13  Angelus
```