LIA - DEIS - Facoltà di Ingegneria - Università di Bologna

Elementi di informatica LB

Corso di laurea in Ingegneria Elettrica - Anno Accademico 2007-2008

Progetti su più file su tabelle

Esercizio n.1.7

È dato un file di testo **PEOPLE.TXT** che contiene i dati di una serie di persone (non più di 20), una persona per riga. Più precisamente, ogni riga contiene nell'ordine:

- il cognome (non più di 20 caratteri, *senza spazi intermedi*)
- uno o più spazi
- il nome (non più di 20 caratteri, *senza spazi intermedi*)
- uno o più spazi
- la data di nascita nel formato **gg/mm/aaaa**
- uno e un solo spazio
- un carattere ('M' o 'F') che indica il sesso.

Si chiede di scrivere un programma C che, dopo aver definito una *struttura* **persona** nel modo appropriato a quanto sopra:
1. contenga una funzione **lettura()** che, dato il nome del file (ed eventualmente altri parametri se opportuno), legga i dati delle persone dal file e li metta in un *array di* ***persona*** di nome **elenco**;
[si mostri a video l'array così costruito]
2. contenga una funzione **compatibili()** che, date due persone, restituisca *vero* se le due persone sono compatibili (intendendo con questo che esse sono di sesso diverso e la differenza di età, *riferita solo all'anno*, non supera i 5 anni), o *falso* altrimenti;
[si mostri a video un esempio d'uso della funzione con due persone scelte all'array a vostro piacere]
3. chieda all'utente, da console, cognome, nome, sesso e data di nascita
4. in base a quanto ottenuto al punto precedente, scriva i dati delle persone compatibili con l'utente sia a video sia su un file binario di nome **PARTNERS.DAT**.

Soluzione

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

#define NUMEROPERSONE 20
#define DIMCOGNOME 21
#define DIMNOME 21
```

```
struct data {
    int giorno, mese, anno;
};

struct persona {
    char cognome[DIMCOGNOME], nome[DIMNOME], sesso;
    struct data nascita;
};



/* ---------- domanda 1 -------- */

void lettura(char nomefile[], struct persona v[], int* pindice){
  struct persona x;
  FILE *f = fopen(nomefile, "r");
  if (f==NULL) {
    printf("Impossibile aprire file di ingresso");
    exit(1); } /* se non riesce a creare il file
                  visualizza un messaggio di errore ed esce dal programma */

  while (fscanf(f, "%s%s%d/%d/%d %c\n", x.cognome, x.nome,
                  &x.nascita.giorno, &x.nascita.mese, &x.nascita.anno,
                     &x.sesso)>0) {
     v[*pindice] = x;
     (*pindice)++;
  }
  fclose(f);
}

void mostraElenco(struct persona elenco[], int dim){
  int i;
  for (i=0; i<dim; i++)
         printf("%s %s nato(a) il %d/%d/%d\n", elenco[i].nome,
                  elenco[i].cognome,elenco[i].nascita.giorno,
                   elenco[i].nascita.mese, elenco[i].nascita.anno);
}

/* ---------- domanda 2 -------- */

int compatibili(struct persona p1, struct persona p2){
    int comp_sesso,comp_anno;
    comp_sesso=p1.sesso!=p2.sesso;
    if (abs(p1.nascita.anno-p2.nascita.anno)<=5) comp_anno=1;
    else comp_anno=0;
    return comp_sesso && comp_anno;
}

/* ---------- domande 3 e 4 -------- */

main(){
  struct persona elenco[NUMEROPERSONE], utente;
  int indiceElenco = 0, i;
  FILE *fbin;
  /* ---------------- prova domanda 1 --------------------*/
  lettura("PEOPLE.TXT", elenco, &indiceElenco);
  mostraElenco(elenco, indiceElenco);
  /* ---------------- prova domanda 2 --------------------*/
  printf("Le due persone %s %s e %s %s sono %s\n",
    elenco[0].nome, elenco[0].cognome, elenco[1].nome, elenco[1].cognome,
    compatibili(elenco[0],elenco[1]) ? "compatibili" : "incompatibili");
```

```
  /* -------------------- domanda 3 ----------------------*/

  printf("\n\nInserire il proprio nome e cognome: ");
  scanf("%s%s", utente.nome, utente.cognome);
  printf("Inserire la propria data di nascita (gg/mm/aaaa): ");
  scanf("%d/%d/%d", &utente.nascita.giorno, &utente.nascita.mese,
                          &utente.nascita.anno);
  scanf("%*c"); /* sopprime il fine linea rimasto sull'input */
  printf("Inserire il sesso (M/F): ");
  scanf("%c%*c",&utente.sesso);

  /* ------------------- domanda 4 ------------------------*/

  fbin = fopen("PARTNERS.DAT", "wb");
  if (fbin==NULL) {
    printf("Impossibile aprire file di uscita\n");
    exit(2); }
    /* se non riesce a creare il file
    visualizza un messaggio di errore ed esce dal programma */
  printf("\nElenco persone compatibili con %s %s:\n",
            utente.nome, utente.cognome);
  for (i=0; i<indiceElenco; i++)
     if (compatibili(utente,elenco[i])) {
         fwrite(&elenco[i], sizeof(struct persona), 1, fbin);
         printf("%s %s nato(a) il %d/%d/%d\n", elenco[i].nome,
                    elenco[i].cognome, elenco[i].nascita.giorno,
                  elenco[i].nascita.mese, elenco[i].nascita.anno);
  }
  fclose(fbin);
}
```

Esercizio n.3

Si scriva un programma in Linguaggio C che conta il numero di 'A','a' e 'B','b' in un file il cui nome viene letto da input.

```
#include <stdio.h>

void conta01(char *nomefile, int *pconta0, int *pconta1)
  /* Conta il numero di '0' e '1' presenti nel file di nome nomefile e
     restituisce il risultato del conteggio in *pconta0 e *pconta1. */
{
  FILE *fp;
  char ch;

  if ((fp = fopen(nomefile, "r")) == NULL) {
    printf("Errore aprendo in lettura il file %s\n", nomefile);
    exit(1);
  }

  *pconta0 = 0;
  *pconta1 = 0;
  while ((ch = fgetc(fp)) != EOF) {
    switch (ch) {
    case '0': (*pconta0)++; break;
    case '1': (*pconta1)++; break;
```

```
    }
  }

  fclose(fp);
} /* conta01 */


int main(void)
{
  FILE *ptrFile;
  int conta0, conta1;
  char nomefile[257];

  printf("Introdurre il nome del file: ");
  scanf("%12s", nomefile);

  conta01(nomefile, &conta0, &conta1);

  printf("Il numero di 0 nel file e`: %d\n", conta0);
  printf("Il numero di 1 nel file e`: %d\n", conta1);

  return 0;
}
```

Esercizio n.4

Si scriva un progetto su più file in Linguaggio C in cui preso un file di testo il cui nome venga passato per input lo si ricopi in un file di testo diverso criptando il contenuto attraverso la cosiddetta conversione carbonari

```
#include <stdio.h>

void ConvertiFile(char *filein, char *fileout);

int main(void)
{
  char nomefileinput[256], nomefileoutput[256];

  printf("Introdurre il nome del file da convertire: ");
  scanf("%12s", nomefileinput);
  printf("Introdurre il nome del file dopo la conversione: ");
  scanf("%12s", nomefileoutput);
  ConvertiFile(nomefileinput, nomefileoutput);

  return 0;
}

void ConvertiFile(char *filein, char *fileout)
{
  FILE *ptrfilein, *ptrfileout;
  int car;

  if ((ptrfilein = fopen(filein, "r")) == NULL) {
    printf("Non e` possibile aprire in lettura il file %s\n", filein);
    exit(1);
  }

  if ((ptrfileout = fopen(fileout, "w")) == NULL) {
    printf("Non e` possibile aprire in scrittura il file %s\n", fileout);
    exit(1);
  }

  car = fgetc(ptrfilein);
```

```
  while (!feof(ptrfilein)) {
    switch (car) {
    case 'A': fputc('O', ptrfileout); break;
    case 'O': fputc('A', ptrfileout); break;
    case 'B': fputc('P', ptrfileout); break;
    case 'P': fputc('B', ptrfileout); break;
    case 'C': fputc('G', ptrfileout); break;
    case 'G': fputc('C', ptrfileout); break;
    case 'D': fputc('T', ptrfileout); break;
    case 'T': fputc('D', ptrfileout); break;
    case 'E': fputc('I', ptrfileout); break;
    case 'I': fputc('E', ptrfileout); break;
    case 'F': fputc('V', ptrfileout); break;
    case 'V': fputc('F', ptrfileout); break;
    case 'L': fputc('R', ptrfileout); break;
    case 'R': fputc('L', ptrfileout); break;
    case 'M': fputc('N', ptrfileout); break;
    case 'N': fputc('M', ptrfileout); break;
    case 'S': fputc('Z', ptrfileout); break;
    case 'Z': fputc('S', ptrfileout); break;
    default: fputc(car, ptrfileout); break;
    }
    car = fgetc(ptrfilein);
  }
  fclose(ptrfilein);
  fclose(ptrfileout);
}
```