

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * tema d'esame
 */

/*
Si definisca il tipo Imballo, il cui scopo e' descrivere uno dei colli in giacenza in un ufficio postale. Imballo e' una
struttura con due campi: un array di 20 caratteri Descrizione (che indica il contenuto dell'imballo; un '\0' indica la fine
della stringa) ed un campo float Peso.
Si scriva una funzione, ClassificaImballi, legge da disco il file binario Input, contenente uno o più dati di tipo Imballo,
e produce due nuovi file binari chiamati Pesanti e Leggeri, anch'essi contenenti dati di tipo Imballo. Ciascun dato letto da
Input va copiato in Pesanti oppure in Leggeri a seconda che il suo campo Peso sia maggiore, oppure minore o uguale, di un
valore Soglia, fornito alla funzione tramite un parametro di ingresso. ClassificaImballi crea anche un file di testo,
Rapporto.txt, che contiene una riga con la forma "[ D, P ]" per ciascun dato letto da Input, dove D e' il contenuto del
campo Descrizione e P e' il contenuto del campo Peso.
La funzione ClassificaImballi restituisce il numero di dati letti da Input, oppure EOF se risulta impossibile aprire il file
Input; altri tipi di errore non vanno gestiti. Non e' necessario occuparsi della creazione del file Input.
*/

/* NOTA: i commenti nel programma sono parziali perché si fa riferimento al
 * testo dell'esercizio */

#include <stdio.h>

#define LUNG_DESCR 20
/* lunghezza massima del campo Descrizione del tipo di dato Imballo,
 incluso il '\0' di terminazione */

#define FILE_INPUT    "./file-es11_01/Input"
#define FILE_LEGGERI  "./file-es11_01/Leggeri"
#define FILE_PESANTI  "./file-es11_01/Pesanti"
#define FILE_RAPPORTO "./file-es11_01/Rapporto.txt"
/* i 4 parametri sopra definiti indicano i path dei 4 file di lavoro del
 programma */

typedef struct
{
    char Descrizione[LUNG_DESCR];
```

```
float Peso;  
} Imballo;
```

```
int ClassificaImballi(float Soglia)  
{  
    FILE *pInput, *pPesanti, *pLeggeri;  
    /* identificano gli stream binari */  
    FILE *pRapporto;  
    /* identifica lo stream testuale */  
    FILE *pDestinazione;  
    /* puntatore allo stream binario dove scrivere il dato corrente */  
    Imballo ImbCorrente;  
    /* dato correntemente elaborato (prima letto e poi scritto) */  
    int ContDati;  
    /* conta il numero di elementi letti dal file di input */  
    pInput = fopen(FILE_INPUT, "rb");  
    if ( NULL == pInput )  
    {  
        return(EOF);  
    }  
    pPesanti = fopen(FILE_PESANTI, "wb");  
    pLeggeri = fopen(FILE_LEGGERI, "wb");  
    pRapporto = fopen(FILE_RAPPORTO, "w");  
    ContDati = 0;  
    while (fread(&ImbCorrente, sizeof(Imballo), 1, pInput) > 0)  
    /* finche' ci sono dati nel file di input */  
    {  
        ++ContDati;  
        if (ImbCorrente.Peso > Soglia)  
        {  
            pDestinazione = pPesanti;  
        }  
        else  
        {
```

```
        }
        pDestinazione = pleggeri;
    }
    fwrite(&ImbCorrente, sizeof(Imballo), 1, pDestinazione);
    fprintf(pRapporto, "[ %s, %f ]\n", ImbCorrente.Descrizione,
        ImbCorrente.Peso);
}

fclose(pInput);
fclose(pPesanti);
fclose(pleggeri);
fclose(pRapporto);

return(ContDati);
}

/*****
/***** cio che segue non era richiesto all'esame *****/
/*****/

void StampaFile(char* NomeFile)
/* funzione non richiesta all'esame, usata per debugging: stampa il
 * contenuto di un file binario contenente dati di tipo Imballo.
 * Se il file non esiste o non e' apribile non stampa nulla */
{
    FILE *pDastamp;
    /* identifica lo stream da utilizzare */
    Imballo ImbCorrente;
    /* dato correntemente elaborato */

    pDastamp = fopen(NomeFile, "rb");

    printf("\nIl contenuto del file %s e' il seguente:\n", NomeFile);

    while (fread(&ImbCorrente, sizeof(Imballo), 1, pDastamp))
    /* finche' ci sono dati nel file di input */
    {
        printf("( %s, %f )\n", ImbCorrente.Descrizione, ImbCorrente.Peso);
    }
}
```

```
    }  
    fclose(pDastamp);  
}  
  
int main(void)  
/* il blocco main non era richiesto all'esame. Qui e' usato per verificare  
* il corretto funzionamento di ClassificaImballi. */  
{  
    FILE* pFile;  
    /* identifica lo stream nel quale scrivere i dati */  
    int Cont1, Cont2;  
    /* contatori usati nell'inserimento dei dati nel file di input di prova */  
    Imballo DaInserire;  
    /* elemento da inserire nel file */  
    float Soglia = 2;  
    /* soglia di prova usata nella chiamata a ClassificaImballi */  
    printf("\n\nPer prima cosa verra' creato (o sovrascritto se esistente) il file\n\n\t%\ncontentente dati di prova.\n",  
FILE_INPUT);  
    pFile = fopen(FILE_INPUT, "wb");  
    if ( NULL == pFile )  
    {  
        printf("\nErrore di apertura del file %s!\n", FILE_INPUT);  
    }  
    else  
    {  
        printf("\nIl file %s e' stato correttamente aperto in scrittura.", FILE_INPUT);  
        /* scrittura di alcuni elementi nel file di input */  
        for (Cont1 = 1; Cont1 <= 10; ++Cont1)  
        {  
            /* aggiunta di un elemento al file */  
            DaInserire.Peso = Cont1%2 * 2 + Cont1/3;  
            for (Cont2 = 0; Cont2 <= 18; ++Cont2)
```

```
    {
        DaInserire.Descrizione[Cont2] = 'a' + Cont1 - 1;
    }
    DaInserire.Descrizione[19] = '\0';

    printf("\n-----");
    printf("\nAI file e' stato aggiunto un elemento con queste caratteristiche:");
    printf("\n\tDescrizione\t= \"%s\"", DaInserire.Descrizione);
    printf("\n\tPeso\t\t= %f", DaInserire.Peso);

        fwrite(&DaInserire, sizeof(Imballo), 1, pFile);

        printf("\nDopo questa aggiunta l'indicatore di posizione del file si trova al byte %ld.", ftell(pFile));
        /* NOTA: la stringa di formattazione %ld identifica un long int */
    }
    printf("\n-----");

    printf("\nGenerazione del file di prova %s completata.\nla dimensione totale del file e' di %ld byte.", FILE_INPUT,
    ftell(pFile));

    fclose(pFile);
    printf("\nIl file %s e' stato chiuso.\n", FILE_INPUT);

    StampaFile(FILE_INPUT);

    /* chiamata di ClassificaImballi a scopo di debugging */
    printf("\nla funzione ClassificaImballi (chiamata con soglia %f) ha restituito il valore %d.\n", Soglia,
    ClassificaImballi(Soglia));

    StampaFile(FILE_PESANTI);
    StampaFile(FILE_LEGGERI);
}

printf("\nIl contenuto del file di testo %s puo' essere ispezionato con qualunque editor di testo.\n\n", FILE_RAPPORTO);
return(0);
}
```

