

Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali
Prof. Penzo
Soluzione compito del 17/02/2012

Esercizio 1

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

```
#define MAX_CLIENTI 15000
```

```
typedef enum{false, true} boolean;
```

```
typedef struct{
    int codiceFC;
    char nome[15];
    char cognome[15];
    int punti;
} cliente;
```

```
typedef struct{
    int codice;
    int codiceCliente;
    char prodotti[15];
    float importo;
} acquisto;
```

// prototipi delle funzioni

```
void aggiungiPunti(cliente C[], int numC, int codiceC, int n);
```

```
int calcolaPunti(acquisto a);
```

```
void aggiornaPuntiCliente(char fileAcquisti[], cliente C[], int numC, int codiceC, int  
*puntiAggiornati);
```

```
void percentualeImportiSuperiori(char fileAcquisti[], float importo, float *perc);
```

// funzione di utilità

```
int puntiCliente(cliente C[], int numC, int codiceC);
```

```
void main(){
    int codiceCliente, punti;
    cliente CLIENTI[MAX_CLIENTI];
    int numClienti; // contiene il numero di clienti presenti nel vettore

    printf("Inserire il codice del cliente: ");
    scanf("%d",&codiceCliente);
    aggiornaPuntiCliente("ACQUISTI.DAT",CLIENTI,numClienti,codiceCliente,&punti);
    printf("Il numero di punti del cliente e' pari a : %d\n",punti);
}
```

// punto 1)

```
void aggiungiPunti(cliente C[], int numC, int codiceC, int n){
    boolean trovato=false;
    int i=0;
```

```

while(i<numC && !trovato){
    if(C[i].codiceFC == codiceC){
        trovato=true;
        C[i].punti += n;
    }
    i++;
}
if(!trovato)
    printf("Errore: cliente non trovato!\n");
}

```

// punto 2)

```

int calcolaPunti(acquisto a){
    FILE *fp;
    int codiceP, punti, totPunti=0;
    char descr[30];
    float costo;

    fp = fopen(a.prodotti,"rt");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        fscanf(fp,"%d%s%f%d",&codiceP,descr,&costo,&punti);
        while(!feof(fp)){
            totPunti += punti;
            fscanf(fp,"%d%s%f%d",&codiceP,descr,&costo,&punti);
        }
        fclose(fp);
    }
    return totPunti;
}

```

// punto 3)

```

void aggiornaPuntiCliente(char fileAcquisti[], cliente C[], int numC, int codiceC, int
*puntiAggiornati){
    FILE *fp;
    acquisto a;
    int totPunti=0;

    fp = fopen(fileAcquisti,"rb");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        fread(&a,sizeof(acquisto),1,fp);
        while(!feof(fp)){
            if(a.codiceCliente==codiceC)
                totPunti += calcolaPunti(a);
            fread(&a,sizeof(acquisto),1,fp);
        }
        fclose(fp);
        aggiungiPunti(C,numC,codiceC,totPunti);
    }
}

```

```

        *puntiAggiornati = puntiCliente(C,numC,codiceC);
    }
}

```

// funzione di utilità

```

int puntiCliente(cliente C[], int numC, int codiceC){
    boolean trovato=false;
    int i=0, punti=0;;

    while(i<numC && !trovato){
        if(C[i].codiceFC==codiceC){
            trovato=true;
            punti = C[i].punti;
        }
        i++;
    }
    return punti;
}

```

// punto 4)

```

void percentualeImportiSuperiori(char fileAcquisti[], float importo, float *perc){
    FILE *fp;
    acquisto a;
    int totAcquisti=0, totSuperiori=0;

    fp=fopen(fileAcquisti,"rb");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        fread(&a,sizeof(acquisto),1,fp);
        while(!feof(fp)){
            totAcquisti++;
            if(a.importo >= importo)
                totSuperiori++;
            fread(&a,sizeof(acquisto),1,fp);
        }
        fclose(fp);
    }
    if(totAcquisti>0)
        *perc = totSuperiori/(float)totAcquisti*100;
    else
        *perc = 0;
}

```