

Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali
Prof. Penzo
Soluzione compito del 18/02/2011

Esercizio 1

```
#include<stdio.h>
#include<string.h>
#include<math.h>
#define MAX_CAR_PER_RIGA 20
#define MAX_RIGHE 5
#define MAX_CLIENTI 10000
#define MAX_NOTIZIE 20
#define LUNGHEZZA_CELL 11
#define COSTO_NOTIZIA 20

typedef enum {sport, politica, cronaca} tipoNotizia;
typedef enum {false, true} boolean;
typedef char riga[MAX_CAR_PER_RIGA];

typedef struct{
    tipoNotizia tipo;
    riga testo[MAX_RIGHE];
    int numRighe;
} notizia;

typedef struct{
    char cellulare[LUNGHEZZA_CELL];
    tipoNotizia interesse;
    int maxNotizie;
    float credito;
} cliente;

// prototipi delle funzioni
void inserisciCliente(cliente C[], int *numC, cliente c);
int notiziePerTipo(char fileNotizie[], tipoNotizia t, notizia N[], int maxN);
void notiziePerCliente(cliente c, char fileNotizie[], char fileOutput[], float *costo);
boolean invioNotizieACliente(char cell[], cliente C[], int numC, char fileOutput[]);

void main(){
    cliente CLIENTI[MAX_CLIENTI];
    int numClienti; // contiene il numero corrente di clienti nel vettore
    char cell[LUNGHEZZA_CELL];
    char filename[20];
    boolean esito;

    // punto a)
    printf("Inserire il numero di cellulare: ");
    scanf("%s",cell);
    printf("Inserire il nome del file delle notizie per il cliente");
    scanf("%s",filename);
    esito = invioNotizieACliente(cell,CLIENTI,numClienti,filename);
```

```

        if(esito==true)
            printf("Credito aggiornato correttamente\n");
        else
            printf("Invio notizie non riuscito");

    }

```

// punto 1)

```

void inserisciCliente(cliente C[], int *numC, cliente c){

    if(*numC<MAX_CLIENTI){
        strcpy(C[*numC].cellulare,c.cellulare);
        C[*numC].interesse = c.interesse;
        C[*numC].maxNotizie = c.maxNotizie;
        C[*numC].credito = c.credito;
        (*numC)++;
    }
}

```

// punto 2)

```

int notiziePerTipo(char fileNotizie[], tipoNotizia t, notizia N[], int maxN){
    int numNotizie=0, i;
    FILE *fp;
    notizia n;
    boolean completo=false;

    fp = fopen(fileNotizie,"rb");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        fread(&n,sizeof(notizia),1,fp);
        while(!feof(fp) && !completo){
            if(n.tipo == t && numNotizie < maxN){
                N[numNotizie].tipo=n.tipo;
                for(i=0;i<n.numRighe;i++)
                    strcpy(N[numNotizie].testo[i],n.testo[i]);
                numNotizie++;
            }
            else if(numNotizie == maxN){
                completo=true;
                printf("Errore: il vettore delle notizie e' completo!\n");
            }
            fread(&n,sizeof(notizia),1,fp);
        }
        fclose(fp);
    }
    return(numNotizie);
}

```

// punto 3)

```
void notiziePerCliente(cliente c, char fileNotizie[], char fileOutput[], float *costoTotale){

    int numNotizie, maxNotiziePerCredito, i, j;
    notizia Notizie[MAX_NOTIZIE];
    FILE *fp;

    numNotizie = notiziePerTipo(fileNotizie,c.interesse,Notizie,MAX_NOTIZIE);
    fp = fopen(fileOutput,"w");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        maxNotiziePerCredito = floor(c.credito*100/COSTO_NOTIZIA);
        for(i=0; i<numNotizie && i<c.maxNotizie && i<maxNotiziePerCredito; i++){
            for(j=0; j<Notizie[i].numRighe; j++){
                fputs(Notizie[i].testo[j],fp);
                fputs("\n",fp);
            }
            *costoTotale = i*COSTO_NOTIZIA/100;
            fclose(fp);
        }
    }
}
```

// punto 4)

```
boolean invioNotizieACliente(char cell[], cliente C[], int numC, char fileOutput[]){

    boolean trovato=false, aggiornatoCredito=false;
    float costoTotale;
    riga r;
    FILE *fp;
    int i=0;

    while(i<numC && !trovato)
        if(!strcmp(C[i].cellulare,cell))
            trovato=true;
        else
            i++;

    if(trovato){
        notiziePerCliente(C[i],"NOTIZIE.DAT",fileOutput,&costoTotale);
        fp = fopen(fileOutput,"r");
        if(fp==NULL)
            printf("Errore di apertura file!\n");
        else{
            fgets(r,1,fp);
            while(!feof(fp)){
                puts(r);
                fgets(r,1,fp);
            }
            fclose(fp);
        }
    }
}
```

```
        C[i].credito -= costoTotale;
        aggiornatoCredito=true;
    }
    else
        printf("Errore: cliente non trovato!\n");
    return aggiornatoCredito;
}
```