

Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali
Prof. Penzo
Soluzione compito del 09/10/2009

Esercizio 1

```
#include<stdio.h>
#include<string.h>
#define MAX_PRIMAPAGINA 20
#define MAX_NUMERI 366

typedef enum{politica, cronaca, sport} tipoArticolo;
typedef enum{false, true} boolean;

typedef struct{
    int codice;
    char titolo[60];
    char autore[30];
    tipoArticolo tipo;
    char dataPubblicazione[11];
} articolo;

typedef struct{
    int codice;
    char dataUscita[11];
    int primaPagina[MAX_PRIMAPAGINA];
    int numArticoliPrimaPagina;
    int numCopie;
} numeroQuotidiano;

// prototipi delle funzioni
int massimoNumeroCopie(numeroQuotidiano NQ[], int numNQ, char data[]);
tipoArticolo tipoArticoloPiuFrequente(char fileArticoli[]);
void cercaDatiArticolo(int codArticolo, char fileArticoli[], char titolo[], char autore[]);
void generaPrimaPagina(numeroQuotidiano NQ[], int numNQ, char dataUscita[], char
filePrimaPagina[], char fileArticoli[]);

void main(){

    tipoArticolo t;

    t = tipoArticoloPiuFrequente("ARTICOLI.DAT");
    printf("Il tipo di articolo piu' frequente e': ");
    if(t == politica)
        printf("politica\n");
    else if(t == cronaca)
        printf("cronaca\n");
    else
        printf("sport\n");
}
```

// punto 1)

```
int massimoNumeroCopie(numeroQuotidiano NQ[], int numNQ, char data[]){

    int maxNumeroCopie, indiceMaxNumeroCopie=0, i;

    for (i=1; i<numNQ; i++)
        if(NQ[i].numCopie > NQ[indiceMaxNumeroCopie].numCopie)
            indiceMaxNumeroCopie = i;
    maxNumeroCopie = NQ[indiceMaxNumeroCopie].numCopie;
    strcpy(data,NQ[indiceMaxNumeroCopie].dataUscita);
    return(maxNumeroCopie);
}
```

// punto 2)

```
tipoArticolo tipoArticoloPiuFrequente(char fileArticoli[]){

    FILE *fp;
    articolo a;
    int numPolitica=0, numCronaca=0, numSport=0;

    fp = fopen(fileArticoli,"rb");
    if(fp==NULL)
        printf("Errore di apertura file!\n");
    else{
        fread(&a,sizeof(articolo),1,fp);
        while(!feof(fp)){
            if(a.tipo==politica)
                numPolitica++;
            else if(a.tipo==cronaca)
                numCronaca++;
            else
                numSport++;
            fread(&a,sizeof(articolo),1,fp);
        }
        fclose(fp);
    }
    if(numPolitica > numCronaca && numPolitica > numSport)
        return(politica);
    else if (numCronaca > numPolitica && numCronaca > numSport)
        return(cronaca);
    else
        return(sport);
}
```

// punto 3)

```
void cercaDatiArticolo(int codArticolo, char fileArticoli[], char titolo[], char autore[]){

    boolean trovato = false;
    FILE *fp;
    articolo a;
```

```

fp = fopen(fileArticoli,"rb");
if(fp==NULL)
    printf("Errore di apertura file!\n");
else{
    fread(&a,sizeof(articolo),1,fp);
    while(!feof(fp) && !trovato){
        if(a.codice == codArticolo){
            trovato = true;
            strcpy(titolo, a.titolo);
            strcpy-autore, a-autore);
        }
        fread(&a,sizeof(articolo),1,fp);
    }
    fclose(fp);
    if(!trovato)
        printf("Errore, articolo non trovato!\n");
}
}

```

// punto 4)

```

void generaPrimaPagina(numeroQuotidiano NQ[], int numNQ, char dataUscita[], char
filePrimaPagina[],char fileArticoli[]){

    boolean trovato = false;
    int i=0, indiceTrovato;
    char titolo[60], autore[30];
    FILE *fp;

    while(i < numNQ && !trovato){
        if(!strcmp(NQ[i].dataUscita,dataUscita)){
            trovato=true;
            indiceTrovato = i;
        }
        i++;
    }
    if(!trovato)
        printf("Errore, numero quotidiano non trovato!\n");
    else{
        fp = fopen(filePrimaPagina,"w");
        if(fp==NULL)
            printf("Errore di apertura file!\n");
        else{
            for(i=0; i < NQ[indiceTrovato].numArticoliPrimaPagina; i++){

                cercaDatiArticolo(NQ[indiceTrovato].primaPagina[i], fileArticoli, titolo, autore);
                fprintf(fp, "Titolo: %s\t Autore: %s\n",titolo, autore);
            }
            fclose(fp);
        }
    }
}

```