

Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali
Prof. Penzo
Soluzione compito del 12/02/2010

Esercizio 1

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

```
#define MAX_IMPIANTI 500
#define MAX_REGIONI 20
```

```
typedef enum{ fotovoltaico, idroelettrico, geotermico, eolico } tipoImpianto;
typedef enum{ false, true } boolean;
```

```
typedef char tipoRegione[20];
```

```
typedef struct{
    int codice;
    tipoImpianto tipologia;
    int potenzaProdotta;
    tipoRegione regioni[MAX_REGIONI];
    int numRegioni;
} impianto;
```

```
typedef struct{
    int codiceEdificio;
    char regione[25];
    int codiceImpianto;
    int annoInizioFornitura;
} edificiPerTipo;
```

// prototipi delle funzioni

```
void numImpiantiPerTipologiaEPotenza(impianto I[], int numI, tipoImpianto tipo, int potenza, int *totImpianti);
```

```
int impiantiFornitori(char fileEdifici[], char regione[], int anno);
```

```
int numImpiantiPerRegione(impianto I[], int numI, char regione[]);
```

```
void edificiPerTipologia(char fileEdifici[], impianto I[], int numI, tipoImpianto tipo, char fileEdificiPerTipo[], int *numRecord);
```

```
void main(){
    impianto IMPIANTI[MAX_IMPIANTI];
    int numImpianti; // numero di impianti presenti correntemente nel vettore
    char tipologia[15], nomeFile[20];
    int numR;
    tipoImpianto tipo;
```

// punto a)

```
printf("Inserire la tipologia di impianto: ");
```

```
gets(tipologia); // si assume che l'utente inserisca un valore valido di tipologia di impianto
```

```
if(!strcmp(tipologia,"fotovoltaico"))
```

```
    tipo = fotovoltaico;
```

```

else if(!strcmp(tipologia,"idroelettrico"))
    tipo = idroelettrico;
else if(!strcmp(tipologia,"geotermico"))
    tipo = geotermico;
else if(!strcmp(tipologia,"eolico"))
    tipo = eolico;
printf("Inserire il nome del file da generare: ");
gets(nomeFile);
edificiPerTipologia("EDIFICI.TXT",IMPIANTI,numImpianti,tipo,nomeFile,&numR);
printf("Il numero di record riportati nel file e': %d\n",numR);
}

```

// punto 1)

```

void numImpiantiPerTipologiaEPotenza(impianto I[], int numI, tipoImpianto tipo, int potenza, int
*totImpianti){

```

```

int i;

```

```

*totImpianti=0;
for(i=0; i<numI; i++)
    if(I[i].tipologia == tipo && I[i].potenzaProdotta >= potenza)
        (*totImpianti)++;
}

```

// punto 2)

```

int impiantiFornitori(char fileEdifici[], char regione[], int anno){

```

```

FILE *fp;

```

```

int codiceEdificio, codiceImpianto, annoInizio, numCodici=0;
char regioneEdificio[25];

```

```

fp = fopen(fileEdifici,"rt");

```

```

if(fp==NULL)

```

```

    printf("Errore di apertura file!\n");

```

```

else{

```

```

    fscanf(fp,"%d%s%d%d",&codiceEdificio,regioneEdificio,&codiceImpianto,&annoInizio);

```

```

    while(!feof(fp)){

```

```

        if(!strcmp(regioneEdificio,regione) && annoInizio > anno){

```

```

            printf("Codice impianto: %d\n",codiceImpianto);

```

```

            numCodici++;

```

```

        }

```

```

        fscanf(fp,"%d%s%d%d",&codiceEdificio,regioneEdificio,&codiceImpianto,&annoInizio);

```

```

    }

```

```

    fclose(fp);

```

```

}

```

```

return(numCodici);

```

```

}

```

// punto 3)

```

int numImpiantiPerRegione(impianto I[], int numI, char regione[]){

```

```

int i, j, numImpianti=0;
boolean trovataRegione;
for(i=0; i<numI; i++){
    trovataRegione=false;
    for(j=0; j<I[i].numRegioni && !trovataRegione; j++){
        if(!strcmp(I[i].regioni[j], regione)){
            trovataRegione=true;
            numImpianti++;
        }
    }
}
return(numImpianti);
}

```

// punto 4)

```

void edificiPerTipologia(char fileEdifici[], impianto I[], int numI, tipoImpianto tipo, char
fileEdificiPerTipo[], int *numRecord){

```

```

FILE *fpI, *fpO;
int i,codiceEdificio,codiceImpianto,annoInizio;
char regione[25];
boolean trovatoImpianto;
edificiPerTipo e;

*numRecord=0;
fpI = fopen(fileEdifici,"rt");
fpO = fopen(fileEdificiPerTipo,"wb");
if(fpI==NULL || fpO==NULL)
    printf("Errore di apertura file!\n");
else{
    fscanf(fpI,"%d%s%d%d",&codiceEdificio,regione,&codiceImpianto,&annoInizio);
    while(!feof(fpI)){
        trovatoImpianto = false;
        i=0;
        while(i<numI && !trovatoImpianto){
            if(I[i].codice == codiceImpianto){
                trovatoImpianto = true;
                if(I[i].tipologia == tipo)
                    e.codiceEdificio = codiceEdificio;
                    e.codiceImpianto = codiceImpianto;
                    e.annoInizioFornitura = annoInizio;
                    fwrite(&e,sizeof(edificiPerTipo),1,fpO);
                    (*numRecord)++;
            }
            i++;
        }
        fscanf(fpI,"%d%s%d%d",&codiceEdificio,regione,&codiceImpianto,&annoInizio);
    }
    fclose(fpI);
    fclose(fpO);
}}

```