

```
/* Versione C del compito Java del 16/09/2002 */
```

```
/*
```

Si scriva un programma C per mantenere informazioni relative a 10 pizzerie. Le pizzerie devono essere memorizzate su un vettore.

Ogni pizzeria possiede le seguenti informazioni:

- nome
- indirizzo
- telefono
- giorno di chiusura (stringa)
- menu di pizze

Ogni menu di pizze e' costituito da un vettore di 30 pizze.

Ogni pizza e' costituita dalle seguenti informazioni:

- nome
- ingredienti (vettore di max 10 ingredienti di tipo stringa)
- prezzo

Si scrivano le definizioni delle seguenti funzioni C:

1) aggiungiPizza: dato un menu e una pizza, inserisce la pizza nel menu

2) getGiornoDiChiusura: data una pizzeria, restituisce il giorno di chiusura

3) setGiornoDiChiusura: data una pizzeria e un giorno, modifica il giorno di chiusura di quella pizzeria

4) inserisciNuovaPizza: data una pizzeria e una pizza, inserisce la pizza nel menu della pizzeria

5) trovaPizzeSenzaIngrediente: dato un menu e un ingrediente, restituisce un vettore di pizze, contenente le pizze nel menu che non contengono quell'ingrediente.

Il menu deve rimanere invariato. La funzione deve anche restituire il numero di pizze senza ingrediente trovate.

```
*/
```

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

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

```
#define MAXINGREDIENTI 10
```

```
#define MAXPIZZE 30
```

```
#define MAXPIZZERIE 10
```

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

```
typedef char ingrediente[20];
```

```
typedef struct{
    char nome[20];
    ingrediente ingredienti[MAXINGREDIENTI];
    int numIngredienti;
    int prezzo;
}
```

```
    }pizza;
```

```
typedef struct{
    char via[30];
    int numerocivico;
    int cap;
    char citta[20];
}
```

```
    }address;
```

```
typedef struct{
    char prefisso[5];
    int numero;
}
```

```
    }phone;
```

```
typedef struct{
    char nome[20];
    address indirizzo;
    phone telefono;
    char giornoDiChiusura[10];
    pizza menu[MAXPIZZE];
    int numPizze;
}
```

```
    }pizzeria;
```

```
void aggiungiPizza(pizza menu[], int *numpizze, pizza p);
```

```
void getGiornoDiChiusura(pizzeria p, char giorno[]);
void setGiornoDiChiusura(pizzeria *p, char giorno[]);
void inserisciNuovaPizza(pizzeria *p, pizza pz);
int trovaPizzeSenzaIngrediente(pizza menu[], int numpizze, ingrediente i, pizza
senzaIngrediente[]);

main(){

pizzeria VetPizzerie[MAXPIZZERIE], p;
pizza senzaIngrediente[MAXPIZZE], nuovapizza;
int numingr,numSenza,i;
char chiusura[10];
ingrediente ingr;

printf("Inserisci i dati della prima pizzeria:\n");
printf("Nome: ");
gets(p.nome);
printf("Indirizzo:\n");
printf("Via: ");
gets(p.indirizzo.via);
printf("Numero: ");
scanf("%d",&(p.indirizzo.numerocivico));
printf("CAP: ");
scanf("%d",&(p.indirizzo.cap));
printf("Citta': ");
getchar();
gets(p.indirizzo.citta);
printf("Prefisso: ");
scanf("%s",p.telefono.prefisso);
printf("Numero: ");
scanf("%d",&(p.telefono.numero));
printf("Giorno di chiusura: ");
scanf("%s",p.giornoDiChiusura);
p.numPizze=0;

// inserisciPizzeria(VetPizzerie,&numpizzerie,p);

printf("Inserisci i dati di una pizza: \n");
printf("Nome: ");
getchar();
gets(nuovapizza.nome);
printf("Quanti ingredienti? \n");
scanf("%d",&numingr);
getchar();
for(i=0;i<numingr;i++){
    printf("Ingrediente %d: ",i+1);
    gets(nuovapizza.ingredienti[i]);
}

nuovapizza.numIngredienti=i;
printf("Prezzo: ");
scanf("%d",&(nuovapizza.prezzo));

// aggiungiPizza(p.menu,&(p.numPizze),nuovapizza);
inserisciNuovaPizza(&p,nuovapizza);
printf("In giorno di chiusura e': ");
getGiornoDiChiusura(p,chiusura);
printf("%s\n",chiusura);

printf("Inserisci il nuovo giorno di chiusura: ");
getchar();
gets(chiusura);
setGiornoDiChiusura(&p,chiusura);
```

```
printf("Il nuovo giorno di chiusura e': ");
getGiornoDiChiusura(p, chiusura);
printf("%s\n", chiusura);

printf("Inserisci l'ingrediente da ricercare: ");
gets(ingr);
numSenza=trovaPizzeSenzaIngrediente(p.menu, p.numPizze, ingr, senzaIngrediente);
printf("Le pizze senza ingrediente sono: %d\n", numSenza);

}

void aggiungiPizza(pizza menu[], int *numpizze, pizza p){
    if(*numpizze<MAXPIZZE){
        menu[*numpizze]=p; // qui copiare elemento per elemento
        (*numpizze)++;
    }
    else
        printf("Non c'e' spazio per la pizza!");
}

void getGiornoDiChiusura(pizzeria p, char giorno[]){
    strcpy(giorno, p.giornoDiChiusura);
}

void setGiornoDiChiusura(pizzeria *p, char giorno[]){
    strcpy((*p).giornoDiChiusura, giorno);
}

void inserisciNuovaPizza(pizzeria *p, pizza pz){
    aggiungiPizza((*p).menu, &((*p).numPizze), pz);
}

boolean contieneIngrediente(pizza p, ingrediente ingr){
    int i=0;
    boolean trovato=false;
    while(i<p.numIngredienti && !trovato){
        if(!strcmp(p.ingredienti[i], ingr))
            trovato=true;
        i++;
    }
    return(trovato);
}

int trovaPizzeSenzaIngrediente(pizza menu[], int numPizze, ingrediente ingr, pizza senzaIngrediente[]){
    int i, numPizzeSenza=0;
    for(i=0; i<numPizze; i++){
        if(!contieneIngrediente(menu[i], ingr)){
            senzaIngrediente[numPizzeSenza]=menu[i];
            numPizzeSenza++;
        }
    }
    return(numPizzeSenza);
}
```