

/* Scrivere un programma che legga dallo standard input i dati relativi ad un elenco di utenti telefonici (max 100) e li memorizzi in un vettore.

Ogni elemento dell'archivio e' caratterizzato dalle informazioni:

- nome
- cognome
- telefono (prefisso e numero telefonico)

Una volta inizializzato l'archivio con alcuni dati forniti dall'utente (il numero di record inseriti e' imprecisato e deciso dall'utente), il programma deve essere in grado di attuare varie richieste dell'utente:

- 1) stampa: visualizzazione sullo standard output del contenuto dell'archivio
- 2) ricerca: dati in ingresso nome e cognome di una persona, si richiede la visualizzazione del numero telefonico relativo alla persona data
- 3) aggiornamento: modifica dei dati di un utente. Vengono forniti in ingresso nome, cognome e nuovo numero di telefono di una persona e il programma assegna il nuovo numero alla persona
- 4) inserimento: inserimento di un nuovo record nell'archivio, dati nome, cognome e numero di telefono della persona da inserire
- 5) cancellazione: eliminazione di un elemento dall'archivio, dati il nome e il cognome della persona da cancellare
- 6) uscita: termine del programma

L'interazione tra utente e programma avviene in modo ciclico:
l'utente puo' sottoporre una richiesta ad ogni ciclo.
L'esecuzione del programma termina quando l'utente richiede l'uscita. */

```
#include<stdio.h>
#define MAXUTENTI 4

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

typedef struct{
    char nome[20];
    char cognome[20];
    phone telefono;
}utente;

typedef enum{false,true} Boolean;

main()
{
    utente UTENTI[MAXUTENTI];
    char nome[20],cognome[20],prefisso[5];
    int inseriti,op,i,j,numero;
    Boolean fine,trovato;
    char scelta,c;

    inseriti=0;
    fine=false;
    while(inseriti<MAXUTENTI && !fine){
        printf("Vuoi inserire un utente? ");
        scelta=getchar();
        c=getchar();
        if(scelta=='s' || scelta=='S'){
            printf("Inserisci il nome: ");
            gets(UTENTI[inseriti].nome);
            printf("Inserisci il cognome: ");
            gets(UTENTI[inseriti].cognome);
        }
    }
}
```

```

        printf( "Inserisci il prefisso: " );
        scanf( "%s" , UTENTI[inseriti].telefono.prefisso );
        printf( "Inserisci il numero: " );
        scanf( "%d" , &UTENTI[inseriti].telefono.numero );
        inseriti++;
        c=getchar();
    }
    else fine=true;
}

fine=false;
do{
printf( "\n" );
printf( "Quale operazione vuoi svolgere sull'archivio?\n" );
printf( "1: stampa\n" );
printf( "2: ricerca\n" );
printf( "3: aggiornamento\n" );
printf( "4: inserimento\n" );
printf( "5: cancellazione\n" );
printf( "6: uscita\n" );
scanf( "%d" , &op );
c=getchar();
switch(op){
    case(1): /* stampa */
        for(i=0;i<inseriti;i++){
            printf( "Dati dell'utente %d: \n" , i+1 );
            printf( "Nome: %s\n" , UTENTI[i].nome );
            printf( "Cognome: %s\n" , UTENTI[i].cognome );
            printf( "Prefisso: %s\n" , UTENTI[i].telefono.prefisso );
;
            printf( "Numero: %d\n" , UTENTI[i].telefono.numero );
        }
        break;
    case(2): /* ricerca */
        printf( "Inserisci il dati della persona da ricercare: \n" );
        printf( "Nome: " );
        gets(nome);
        printf( "Cognome: " );
        gets(cognome);
        i=0;
        trovato=false;
        while(i<inseriti && !trovato){
            if(strcmp(UTENTI[i].nome,nome)==0 && strcmp(UTENTI[i].cognome,cognome)==0){
                trovato=true;
                printf( "Il numero di telefono e': (%s) %d\n" , UTENTI[i].telefono.prefisso,UTENTI[i].telefono.numero );
            }
            i++;
        }
        if(!trovato)
            printf( "L'utente ricercato non appartiene all'archivio!\n" );
        break;

    case(3): /* aggiornamento */
        printf( "Inserisci il dati della persona che cambia telefono: \n" );
        printf( "Nome: " );
        gets(nome);
        printf( "Cognome: " );
        gets(cognome);
        printf( "Nuovo numero di telefono:\n" );
        printf( "Prefisso: " );
        scanf( "%s" , prefisso );
        printf( "Numero: " );
        scanf( "%d" , &numero );
        i=0;
}

```

```

        trovato=false;
        while(i<inseriti && !trovato){
            if(strcmp(UTENTI[i].nome,nome)==0 && strcmp(UTENTI[i].cognome,co
gnome)==0){
                trovato=true;
                strcpy(UTENTI[i].telefono.prefisso,prefi
sso);
                UTENTI[i].telefono.numero=numero;
            }
            i++;
        }
        if(!trovato)
            printf( "L'utente ricercato non appartiene all'archivio!\n" );
        break;
    case(4): /* inserimento */
        if(inseriti<MAXUTENTI){
            printf( "Inserisci il dati della persona da inserire: \n" );
            printf( "Nome: " );
            gets(UTENTI[inseriti].nome);
            printf( "Cognome: " );
            gets(UTENTI[inseriti].cognome);
            printf( "Numero di telefono:\n" );
            printf( "Prefisso: " );
            scanf( "%s" ,UTENTI[inseriti].telefono.prefisso);
            printf( "Numero: " );
            scanf( "%d" ,&UTENTI[inseriti].telefono.numero);
            inseriti++;
        }
        else printf( "Non c'e' piu' spazio nell'archivio!\n" );
        break;

    case(5): /* cancellazione */
        printf( "Inserisci il dati della persona da cancellare: \n" );
        printf( "Nome: " );
        gets(nome);
        printf( "Cognome: " );
        gets(cognome);
        i=0;
        trovato=false;
        while(i<inseriti && !trovato){
            if(strcmp(UTENTI[i].nome,nome)==0 && strcmp(UTEN
TI[i].cognome,cognome)==0){
                trovato=true;
                for(j=i+1;j<inseriti;j++){
                    strcpy(UTENTI[j-1].nome,UTENTI[j
].nome);
                    strcpy(UTENTI[j-1].cognome,UTENT
I[j].cognome);
                    strcpy(UTENTI[j-1].telefono.pref
isso,UTENTI[j].telefono.prefisso);
                    UTENTI[j].telefono.numero=UTEN
TI[j].telefono.numero;
                }
                inseriti--;
            }
            i++;
        }
        if(!trovato)
            printf( "L'utente da cancellare non appartiene all'archivio!" );
        break;
    case(6): /* uscita */
        fine=true;
        break;
    default: printf( "Errore!" );
}
while(!fine);

```

{}