

Fondamenti di Informatica - Ing. Civile/Edile - Dott. Penzo
Soluzione compito A - 16/07/2001

Esercizio 1

- 1) C:\??> md \DIR3\DIR4
- 2) C:\DIR2> copy *.* ..\DIR3\DIR4
- 3) C:\DIR2> cd \DIR1
- 4) C:\DIR1> del ..\DIR2*.txt

Esercizio 2

La prima printf stampa il valore restituito dalla funzione F. Il valore stampato è 1 perché il ciclo for più esterno nella funzione F termina quando l'indice i eguaglia il valore 2 (valore del parametro in ingresso k) e il ciclo for più interno termina quando j eguaglia il valore di i. Poiché l'ultimo valore valido per i è 1 (per i che vale 2 si è detto che il ciclo esterno finisce) il ciclo interno termina quando j assume il valore 1 (senza decrementare il valore di j). Il valore restituito dalla funzione è pertanto $i-j$ ossia $2-1=1$. Poiché la matrice M viene passata per indirizzo a F, la funzione la modifica e il ciclo for del main stampa la matrice così modificata, ossia:

```
1    -1    -2
0     1    -1
0     0     1
```

Esercizio 3

```
#include<stdio.h>
#include<string.h>
#define MAX 100
```

```
typedef struct {
    int giorno;
    int mese;
    int anno;
} data;
```

```
typedef struct {
    char destinazione[20];
    char nome[20];
    char cognome[20];
    data partenza;
    data ritorno;
    int partecipanti;
} prenotazione;
```

```
typedef struct {
    int codice;
```

```
char destinazione[20];
int prezzopers;
char operatore[20];
} offerta;
```

```
void NuovaPrenotazione(prenotazione Vet[], int *numelem);
int MiglioreOfferta(char dest[], char nomefile[], char operatore[]);
void CostoViaggio(prenotazione Vet[], int numelem, char nome[], char cognome[]);
```

```
main()
{
prenotazione tmp, V[MAX];
int N, quanti, max, i=0,j;
char nome[20], cognome[20];

/* qui avviene l'inserimento delle prenotazioni nel vettore */
/* inizializzazione non richiesta dal testo */
/* sia N il numero di elementi inseriti */
....
NuovaPrenotazione(V,&N);
while(i<N) {
    quanti = N - i;
    max = quanti - 1;
    for(j=0; j < quanti; j++)
        if(strcmp(V[j].destinazione,V[max].destinazione)>0)
            max = j;
    if (max < quanti -1)
    {
        tmp = V[quanti -1];
        V[quanti - 1] = V[max];
        V[max] = tmp;
    }
    i++;
}
printf("Inserisci il nome del cliente per cui cercare il minor prezzo complessivo:\n");
gets(nome);
printf("Inserisci il cognome del cliente:\n");
gets(cognome);
CostoViaggio(V,N,nome,cognome);
}
```

```
void NuovaPrenotazione(prenotazione Vet[], int *numelem)
{
if ((*numelem) < MAX)
```

```

{
    printf("Inserisci la destinazione del viaggio:\n");
    gets(Vet[*numelem].destinazione);
    printf("Inserisci il nome del cliente:\n");
    gets(Vet[*numelem].nome);
    printf("Inserisci il cognome del cliente:\n");
    gets(Vet[*numelem].cognome);
    printf("Inserisci la data di partenza (giorno, mese, anno):\n");
    scanf("%d %d %d", &Vet[*numelem].partenza.giorno,
&Vet[*numelem].partenza.mese, & Vet[*numelem].partenza.anno);
    printf("Inserisci la data di ritorno (giorno, mese, anno):\n");
    scanf("%d %d %d", &Vet[*numelem].ritorno.giorno,
&Vet[*numelem].ritorno.mese, & Vet[*numelem].ritorno.anno);
    printf("Inserisci il numero di partecipanti al viaggio:\n");
    scanf("%d", &Vet[*numelem].partecipanti);
    (*numelem)++;
}
else printf("Non è possibile effettuare ulteriori prenotazioni. \n");
}

```

```

int MiglioreOfferta(char dest[], char nomefile[], char operatore[])
{
    offerta elem;
    int minorprezzo = 0, trovato = 0;
    FILE *fp;

    fp = fopen(nomefile, "rb");
    if (fp == NULL)
        printf("Errore di apertura file. \n");
    else
    {
        while (!feof(fp) && !trovato)
        {
            fread (&elem, sizeof(offerta),1,fp);
            if (!strcmp(elem.destinazione, dest))
            {
                trovato = 1;
                minorprezzo = elem.prezzopers;
                strcpy(operatore, elem.operatore);
            }
        }
        while (!feof(fp))
        {
            fread(&elem, sizeof(offerta),1,fp);
            if (!strcmp(elem.destinazione, dest) && elem.prezzopers < minorprezzo)
            {

```

```

        minorprezzo = elem.prezzopers;
        strcpy(operatore, elem.operatore);
    }
}
if (!trovato) printf("Non esiste nessuna offerta per la destinazione richiesta.");
fclose(fp);
}
return(minorprezzo);
}

void CostoViaggio(prenotazione Vet[], int numelem, char nome[], char cognome[])
{
    int i = 0, trovato = 0, p, costo;
    char touroperator[20];

    while (i < numelem && !trovato)
    {
        if (!strcmp(Vet[i].nome, nome) && !strcmp(Vet[i].cognome, cognome))
            trovato = 1;
        else i++;
    }
    if (trovato)
    {
        p = MiglioreOfferta(Vet[i].destinazione, "VIAGGLI.DAT", touroperator);
        if (p > 0)
        {
            costo = (Vet[i].partecipanti) * p;
            printf("importo: %d; tour operator: %s; nome e cognome cliente: %s %s;
destinazione: %s.", costo, touroperator, nome, cognome, Vet[i].destinazione);
        }
    }
    else printf("Il cliente non è compreso fra le prenotazioni. \n");
}

```