Matrici, strutture dati

Loris Giulivi – Nicolò Folloni

Matrici

Vettori di vettori

Le matrici, come i vettori, contengono un insieme di variabili dello stesso tipo. A differenza di questi, le celle nelle matrici sono identificate da due indici anziché uno solo.

```c
void main()
{
    int mat[5][5];

    printf("Inserisci in posizione 0,0: ");
    scanf("%d", &mat[0][0]);
}
```

Esercizio 1

ES1: Scrivere un programma che, data una matrice 8×8 rappresentante una scacchiera sulla quale sono presenti un Re ed una Regina, stampi «sì» se la Regina può mangiare il Re, «no» altrimenti.

SUGGERIMENTO: potete indicare il Re con un **1** e la Regina con un **2**, mentre lo **0** può indicare le caselle vuote.

Esercizio 1 - Soluzione

1

```
void main()
{
    int mat[8][8];
    fill_yes(&mat);

    //Find king's position
    int kx=-1,ky=-1;
    int i,j;
    for(i=0;i<8;i++)
    {
        for(j=0;j<8;j++)
        {
            if(mat[i][j]==1)
            {
                kx=j;
                ky=i;
            }
        }
    }

    if(kx<0 || ky<0)
    {
        printf("Non trovo il re.");
        return;
    }
```

2

```
//Check horizontal
for(j=0;j<8;j++)
{
    if(mat[ky][j]==2)
    {
        printf("Si'");
        return;
    }
}

//Check vertical
for(i=0;i<8;i++)
{
    if(mat[i][kx]==2)
    {
        printf("Si'");
        return;
    }
}
```

3

```
//Check diagonal 2/4
for(i=0;i<(ky-kx);i++)
{
    if(mat[(ky+kx)-i][i]==2)
    {
        printf("Si'");
        return;
    }
}

//Check diagonal 1/3
for(i=0;i<8-(ky-kx);i++)
{
    if(mat[(ky-kx)+i][i]==2)
    {
        printf("Si'");
        return;
    }
}

printf("No");
```

Strutture dati

Struct

Una **`struct`** contiene un insieme di variabili, di tipo potenzialmente diverso, la cui struttura è identificata al momento della definizione di tipo.

```
typedef struct studente
{
    int matricola;
    char nome[100];
    char cognome[100];
} studente;
```

Accesso agli attributi

Gli attributi delle strutture dati così definite possono essere letti/modificati con l'ausilio dell'operatore '**.**'

```
void main()
{
    studente a;
    a.matricola = 123456;
    strcpy(a.nome, "Marco");
    strcpy(a.cognome, "Rossi");
}
```

Esercizio 7

ES7: Definire una **`struct`** per contenere i dati relativi ad uno studente ed una per i dati relativi al risultato di un esame per uno studente.

Scrivere un programma che dato un vettore di risultati di esami e uno di studenti, dopo aver chiesto all'utente un numero di matricola, calcola la media degli esami dello studente.

Esercizio 7 - Soluzione

1

```c
typedef struct studente
{
    int matricola;
    char nome[100];
    char cognome[100];
} studente;

typedef struct risultato
{
    int matricola;
    char corso[100];
    int voto;
} risultato;
```

2

```c
void main()
{
    studente studenti[NUM_STUD];
    risultato risultati[NUM_RIS];
    initialize_structs(studenti, risultati);

    int mat, i, studidx=-1;
    printf("Matricola:");
    scanf("%d", &mat);

    for(i=0;i<NUM_STUD;i++)
    {
        if(studenti[i].matricola==mat)
        {
            studidx=i;
            i=NUM_STUD;
        }
    }
    if(studidx==-1)
    {
        printf("Matricola inesistente");
        return;
    }

    int n_es, somma_voti;
    for(i=0;i<NUM_RIS;i++)
    {
        if(risultati[i].matricola==mat && risultati[i].voto>=18)
        {
            n_es++;
            somma_voti += risultati[i].voto;
        }
    }

    if(n_es==0)
    {
        printf("Lo studente non ha voti!");
        return;
    }

    float media = (float)somma_voti/(float)n_es;
    printf("La media di %s %s e': %f",
           studenti[studidx].cognome,
           studenti[studidx].nome,
           media);
```

Esercizio 8

ES7: Definire una **`struct`** per contenere i dati relativi ad uno studente, identificato da nome, cognome e matricola.

Scrivere un programma che, dopo aver richiesto all'utente i dati di un nuovo studente, lo inserisca in un array di studenti in maniera che l'array sia ordinato per matricole crescenti. Il programma deve permettere l'inserimento di un numero arbitrario di studenti.
Alla fine dell'inserimento, il programma stampa tutti i dati inseriti.

Esercizio 8 - Soluzione

1

```c
typedef struct studente
{
    int matricola;
    char nome[100];
    char cognome[100];
} studente;
```

2

```c
void main()
{
    studente studenti[100];
    int inseriti = 0, mat, i, flag_ignore=0, flag_exit=0;
    char nome[100], cognome[100];
    do
    {
        flag_ignore=0;
        printf("Inserisci nuova matricola (0 per uscire): ");
        scanf("%d", &mat);
        if(mat)
        {
            int ins_idx = inseriti;
            for(i=0;i<inseriti;i++)
            {
                if(studenti[i].matricola==mat)
                {
                    printf("Matricola già esistente\n");
                    flag_ignore=1;
                }
                else if(studenti[i].matricola>mat)
                {
                    ins_idx=i;
                }
            }
            if(!flag_ignore)
            {
                printf("Nome: ");
                scanf("%s", nome);
                printf("Cognome: ");
                scanf("%s", cognome);

                for(i=inseriti;i>ins_idx;i--)
                {
                    studenti[i]=studenti[i-1];
                }

                studenti[ins_idx].matricola = mat;
                strcpy(studenti[ins_idx].nome, nome);
                strcpy(studenti[ins_idx].cognome, cognome);
                inseriti++;
            }
        }
        else
        {
            flag_exit=1;
        }
    }while(!flag_exit);

    for(i=0;i<inseriti;i++)
    {
        printf("Matricola: %d\n", studenti[i].matricola);
        printf("Nome: %s\n", studenti[i].nome);
        printf("Cognome: %s\n\n", studenti[i].cognome);
    }
}
```