Politecnico di Milano

Informatica B, AA 2019/2020

Laboratorio 3

28/10/2019

Amarildo Likmeta (*amarildo.likmeta@polimi.it*)

Luca Frittoli (*luca.frittoli@polimi.it*)

Matrici

Problema 1

Si scriva un programma che prenda in input 2 matrici di interi di dimensione 3x3, ne calcoli la somma e la stampi a schermo.

Suggerimento : la somma di due matrici A e B della stessa dimensione si calcola sommando gli elementi nella stessa posizione.

Esempio :

```
1 2 3       5 1 4       6 3 7
4 5 6   +   5 6 12  =   9 11 18
7 8 9       7 2 11      14 10 20
```

Bonus : implementare la soluzione sia stampando direttamente la somma a schermo man mano che la si calcola, sia salvandola in una terza matrice prima di stamparla.

Soluzione

```
#include <stdio.h>
// Dimensione costante delle matrici
#define N 3
int main()
{
    // Definiamo le tre matrici
    int m1[N][N], m2[N][N], m3[N][N];
    // Indici dei cicli
    int i,j;
    // Prendiamo in input le due matrici
    // Per semplicita' lo facciamo con due soli cicli
    for(i=0;i<N;i++){
        for(j=0;j<N;j++){
            printf("Inserisci l'elemento (%d,%d) della prima matrice\n", i,j);
            scanf("%d", &m1[i][j]);
            printf("Inserisci l'elemento (%d,%d) della seconda matrice\n",
i,j);
            scanf("%d", &m2[i][j]);
            // Calcoliamo la somma
            m3[i][j] = m1[i][j] + m2[i][j];
        }
    }
    printf("\nLa matrice somma e'\n");
    // Stampiamo il risultato
```

```
    for(i=0;i<N;i++){
        for(j=0;j<N;j++){
        printf("%d\t", m3[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Problema 2

Si scriva un programma che prenda in input 2 matrici di interi, le qui dimensioni sono richieste all'utente, e calcoli il prodotto matricale. Verificare che le dimensioni delle matrici siano valide.

Soluzione:

```
#include <stdio.h>

int main()
{
  int n1, m1, n2, m2, i, j, k;
  printf("Inserire le dimensioni delle due matrici da moltiplicare!\n");
  scanf("%d %d %d %d",&n1,&m1, &n2, &m2);
  //Verificiamo se le dimensioni delle matrici siano valide per calcolare il
prodotto
  if (m1 != n2){
    printf("Le dimensioni delle due matrici non sono valide.\n");
    return 0;
  }
  //dichiariamo le due matrici ora che sappiamo le dimensioni
  int mat1[n1][m1], mat2[n2][m2], risultato[n1][m2];
  printf("Inserire gli elementi della prima matrice!\n");
  for(i=0;i<n1;i++){
    for(j=0;j<m1;j++){
        scanf("%d", &mat1[i][j]);
    }
  }
  printf("Inserire gli elementi della seconda matrice!\n");
  for(i=0;i<n2;i++){
    for(j=0;j<m2;j++){
        scanf("%d", &mat2[i][j]);
    }
  }
  //calcoliamo il prodotto delle matrici e salviamo il risultato in una terza
matrice
  //e lo stampiamo allo stesso tempo
  printf("La matrice risultate e :\n");
  for(i=0;i<n1;i++){
    for(j=0;j<m2;j++){
        risultato[i][j]=0;
        for(k=0;k<n2;k++){
            risultato[i][j] += mat1[i][k] * mat2[k][j];
        }
        printf("%d ",risultato[i][j]);
     }
    printf("\n");
  }

return 0;
}
```

Stringhe

Problema 3

Si scriva un programma che prende in input una stringa, trova le vocali stampandole insieme al loro indice e infine stampi una stringa composta solo dalle vocali della stringa originale.

Soluzione

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 100
int main()
{
    // Definiamo la stringa
    char s[MAX_LEN], risultato[MAX_LEN];
    int len,i, count_vocali=0;
    printf("Inserisci una stringa: ");
    gets(s);
    // Calcolo la lunghezza della stringa
    len = strlen(s);
    // Cerchiamo le vocali
    for(i=0;i<len;i++){
        if(   s[i] == 'A' || s[i] == 'a'
           || s[i] == 'e' || s[i] == 'E'
           || s[i] == 'i' || s[i] == 'I'
           || s[i] == 'o' || s[i] == 'O'
           || s[i] == 'u' || s[i] == 'U'){
            printf("Vocale %c all'indice %d\n", s[i], i);
            //aggiungiamo la vocale trovata all'indice giusto della strings risultato
            //e aggiorniamo il contatore
            risultato[count_vocali] = s[i];
            count_vocali += 1;
        }
    }

    // IMPORTANTE: aggiungo il terminatore
    printf("Le vocali della stringa sono:\n");
    risultato[count_vocali] = '\0';
    printf("%s",risultato);
    return 0;
}
```

Problema 4

Si scriva un programma che prenda in input una stringa e converte i caratteri minuscoli in maiuscoli e vice-versa.

Soluzione

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 100
int main()
{
    // Definiamo la stringa
    char s[MAX_LEN];
    int len, i;
    printf("Inserisci una stringa: ");
    gets(s);
    // Calcolo la lunghezza della stringa
    len = strlen(s);
    // Cerchiamo le vocali
    for(i=0;i<len;i++){
        //Convertiamo in carattere maiuscolo aggiungengo a codice ascii la
        //differenza tra un carattere maiuscolo e uno minuscolo
        if(s[i]>='a' && s[i] <= 'z')
            s[i] += 'A' - 'a' ;
```

```
        //Convertiamo in carattere minuscolo aggiungengo al codice ascii la
        //differenza tra un carattere minuscolo  e uno maiuscolo
        else if(s[i]>='A' && s[i] <= 'Z')
            s[i] += 'a' - 'A' ;
    }

    printf("La stringa risultate e :\n");
    printf("%s",s);
    return 0;
}
```

Problema 5

Si scriva un programma che richieda all'utente di inserire due stringhe e controlli se la seconda stringa e inclusa nella prima (substring). Nel caso la seconda stringa sia una substringa della prima, stampare l'indice dove si trova.

Soluzione

```
#include <stdio.h>
#define MAX_LEN 100
int main()
{
    char str[MAX_LEN], sub_str[MAX_LEN];
    int len1, len2, indice, i ,j;
    int flag = 0;
    printf("Inserisci una stringa: ");
    gets(str);

    printf("Inserisci la stringa da cercare: ");
    gets(sub_str);

    len1 = strlen(str);
    len2 = strlen(sub_str);

    if (len1 < len2){
        printf("La stringa da cercare e piu lungha della stringa originale!");
        return;
    }
    //Cerchiamo la substringa solo a partire dalle prime <len1 - len2> posizioni
    for(i=0;i<=len1-len2 && flag ==0;i++){
          flag=1;
             for(j=i;j<i+len2 && flag==1;j++){
                    if (str[j]!=sub_str[j-i]){
                flag=0;
                    }
             }
             if (flag==1){
            indice = i;
             }

      }
      if (flag==1)
            printf("\"%s\" si trova nell'indice %d di \"%s\".\n",sub_str, indice, str);
      else
            printf("\"%s\" non si trova in \"%s\".\n", sub_str, str);
    return 0;
}
```

Problema 6

Si scriva un programma che prende in input una stringa (*plaintext*) senza spazi e fatta solo di caratteri maiuscoli, e un numero intero (*key*). Il programma cifra la stringa *plaintext* con il Cifrario di Cesare con chiave *key* , stampando il risultato cifrato (*cyphertext*) a schermo.
Suggerimento : il Cifrario di Cesare è un semplice cifrario a rotazione, che sostituisce una lettera dell'alfabeto con un'altra, ruotandole di *key* posti. Esempio:

key = 3
plaintext = A B C D E F G … X Y Z
cyphertext = D E F G H I J … A B C

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
// Massima lunghezza
#define MAX_LEN 100
int main()
{
char plaintext[MAX_LEN], cyphertext[MAX_LEN];
int key, i;
printf("Inserisci il messaggio da cifrare: ");
// Uso scanf supponendo la stringa non contenga spazi
// Supponiamo inoltre che la stringa inserita contenga solo caratteri
maiuscoli
scanf("%s", plaintext);
printf("Inserisci la chiave: ");
scanf("%d", &key);
// Cifro ogni carattere del plaintext
for(i=0;i<strlen(plaintext);i++)
{
// L'idea della seguente istruzione e':
// - Traslo il carattere i-esimo tra 0 e 25 (sottraendo 'A')
// - Gli sommo la chiave
// - Faccio modulo 26 -> il risultato e' ancora tra 0 e 25
// - Riporto il tutto nell'intervallo originale sommando 'A'
// Esempio: se inserisco Z e chiave 1 ho:
// - Z - A = 25
// - Z - A + 1 = 26
// - (Z - A + 1) % 26 = 0
// - (Z - A + 1) % 26 + 'A' = A
cyphertext[i] = ((plaintext[i] + key - 'A') % 26) + 'A';
}
// Aggiungo il terminatore
cyphertext[i] = '\0';
printf("Il messaggio cifrato e': %s\n",cyphertext);
return 0;
}
```

Problema 7 (Bonus)

Si scriva un programma che permetta di decifrare il seguente messaggio

"NZYRCLEFWLKTZYT"

Il messaggio e' stato codificato con il cifrario di Cesare secondo le regole dell'esercizio precedente.

Suggerimento: dato un qualunque testo, è possibile generare un numero molto piccolo di messaggi cifrati.

Soluzione

```
 #include <stdio.h>
 #include <stdlib.h>
 #include <string.h>
 int main()
 {
 // Il messaggio da decifrare
 char messaggio[20] = "NZYRCLEFWLKTZYT";
 // Una copia che uso per decifrare
 char s[20] = "NZYRCLEFWLKTZYT";
 int len = strlen(messaggio);
 int i;
 int key;
 // Provo tutte le chiavi da 1 a 25
 for(key=1;key<=25;key++)
 {
 // Provo a cifrare il messaggio con key
 for(i=0;i<len;i++)
 {
 s[i] = ((messaggio[i] + key - 'A') % 26) + 'A';
 }
 // Stampo il messaggio cifrato
 // Guardo i messaggi stampati per cercare se qualcuno ha senso...
 printf("%s\n",s);
 }
 return 0;
}
```

Struct & Typedef

Problema 8

Dichiarare un tipo di dato Esame che contenga il nome del corso e il voto ottenuto nel corso. Scrivere poi un programma che prenda in input un Esame e ne stampi i dati.

Soluzione

```
#include <stdio.h>
#define MAX_LEN 100
typedef struct {
    char nome_corso[MAX_LEN];
    float voto_esame;
} Esame;

int main() {
    Esame MioEsame;
    printf("Inserire un Esame\n");

    printf("Nome Corso = ");
    gets(MioEsame.nome_corso);
    printf("Voto = ");
    scanf("%f", &MioEsame.voto_esame);
    printf("\nL'esame letto e': %s con voto %f \n", MioEsame.nome_corso,
MioEsame.voto_esame);
```

```
    return 0;
}
```

Problema 9

Scrivere un programma che usi la struttura dati definita nell'esercizio 8 per prendere in input due Esami, e stampare la media dei voti ottenuti nei due esami.

Soluzione

```
#include <stdio.h>
#define MAX_LEN 100

 typedef struct {
     char nome_corso[MAX_LEN];
     float voto_esame;
 } Esame;

 int main() {
     Esame MieiEsami[2];
     int i;
     float media = 0;
     for(i=0;i<2;i++){
         printf("Inserire Esame %d\n",i+1);
         printf("Nome Corso = ");
         gets(MieiEsami[i].nome_corso);
         printf("Voto = ");
         scanf("%f", &MieiEsami[i].voto_esame);fflush(stdin);
         printf("\nL'esame letto e': %s con voto %f \n",
 MieiEsami[i].nome_corso, MieiEsami[i].voto_esame);
         media += MieiEsami[i].voto_esame;
     }
     printf("Il voto medio e: %f\n", media / 2);

     return 0;
}
```

Problema 10

Dichiarare una struttura dati che usi il tipo di dati Esame degli esercizi precedenti per descrivere uno Studente. Uno studente e descritto da nome e la lista dei esami sostenuti (3 esami). Scrivere poi un programma che prenda in input uno studente e calcoli la media dei suoi voti.

Soluzione

```
 #include <stdio.h>
 #define MAX_LEN 100
 #define N 3
 typedef struct {
     char nome_corso[MAX_LEN];
     float voto_esame;
 } Esame;

 typedef struct {
     char nome_studente[MAX_LEN];
     Esame  voti[N];
 } Studente;

 int main() {
     Studente st;
     int i;
     float media = 0;
     printf("Inserire lo studente\n");
     printf("Nome Studente = ");
     gets(st.nome_studente);
```

```
    for(i=0;i<N;i++){
        printf("Inserire Esame %d\n",i+1);
        printf("Nome Corso = ");
        gets(st.voti[i].nome_corso);
        printf("Voto = ");
        scanf("%f", &st.voti[i].voto_esame);fflush(stdin);
        printf("\nL'esame letto e': %s con voto %f \n", st.voti[i].nome_corso,
st.voti[i].voto_esame);
        media += st.voti[i].voto_esame;
    }
    printf("Il voto medio e: %f\n", media / N);

    return 0;
}
```

Problema 11

Si usi la struttura dati dell'esercizio precedente per definire una classe di M>0 studenti. Si espandi la struttura studente con un campo che contenga la media dei suoi esami.
Si scriva un programma che, preso in input tale classe, effettui due ricerche:
- ricerca dello studente con media dei voti piu alta. Si stampi il nome dello studente e la sua media.
- ricerca dello studente con la differenza piu piccola tra il voto minimo e il voto massimo. Si stampi il nome dello studente, e l'indice e i nomi dei due esami di interesse (voto minimo e voto massimo).

Soluzione

```
#include <stdio.h>
#define MAX_LEN 100
#define N 3
typedef struct {
    char nome_corso[MAX_LEN];
    float voto_esame;
} Esame;

typedef struct {
    char nome_studente[MAX_LEN];
    Esame  voti[N];
    float media_voti;
} Studente;

int main() {
    int M;
    printf("Inserisci il numero di studenti!\n");
    scanf("%d",&M);fflush(stdin);
    Studente classe[M];
    int j,i, studente_media_massima, studente_differenza_minima,
indice_voto_minimo, indice_voto_massimo, ind_min, ind_max;
    float media = 0, media_massima = -1, differenza_minima =100, voto_minimo,
voto_massimo;

    //Leggiamo i dati e calcoliamo allo stesso tempo le due richieste
    for(j=0;j<M;j++){
        printf("Inserire lo studente %d\n", j+1);
        printf("Nome Studente = ");
        gets(classe[j].nome_studente);
        voto_minimo = 100;
        voto_massimo = -1;
        classe[j].media_voti = 0;
        for(i=0;i<N;i++){
            printf("Inserire Esame %d di %s\n",i+1, classe[j].nome_studente);
            printf("Nome Corso = ");
            gets(classe[j].voti[i].nome_corso);
```

```
            printf("Voto = ");
            scanf("%f", &classe[j].voti[i].voto_esame);fflush(stdin);
            if (voto_minimo > classe[j].voti[i].voto_esame){
                voto_minimo = classe[j].voti[i].voto_esame;
                ind_min = i;
            }


            if (voto_massimo < classe[j].voti[i].voto_esame){
                voto_massimo = classe[j].voti[i].voto_esame;
                ind_max = i;
            }

            classe[j].media_voti += classe[j].voti[i].voto_esame;
        }
        if (differenza_minima > voto_massimo - voto_minimo){
            differenza_minima = voto_massimo - voto_minimo;
            studente_differenza_minima = j;
            indice_voto_minimo = ind_min;
            indice_voto_massimo = ind_max;
        }
        classe[j].media_voti /= N;
        if (media_massima < classe[j].media_voti){
            media_massima = classe[j].media_voti;
            studente_media_massima = j;
        }
    }
    printf("Lo studente con media piu alta e %s con media %f e  voti:\n",
classe[studente_media_massima].nome_studente,
classe[studente_media_massima].media_voti);
    for(i=0;i<N;i++){

printf("%s:%f\n",classe[studente_media_massima].voti[i].nome_corso,classe[stude
nte_media_massima].voti[i].voto_esame);
    }
    printf("\n");
    printf("Lo studente con differenza minima di voti e %s con media %f e
voti:\n", classe[studente_differenza_minima].nome_studente,
classe[studente_differenza_minima].media_voti);


        printf("Voto Minimo- Indice %d:%s =%f \n",
indice_voto_minimo,classe[studente_differenza_minima].voti[indice_voto_minimo].
nome_corso,classe[studente_differenza_minima].voti[indice_voto_minimo].voto_esa
me);
       printf("Voto Massimo- Indice %d:%s =%f \n",
indice_voto_massimo,classe[studente_differenza_minima].voti[indice_voto_massimo
].nome_corso,classe[studente_differenza_minima].voti[indice_voto_massimo].voto_
esame);

    }
    return 0;
}
```

Tema d'esame del 02/07/2019 (esercizio 1)

Si vuole sviluppare un programma per analizzare l'andamento di un singolo *titolo societario* in borsa. Il titolo societario ha un nome (ad esempio, IBM oppure AAPL), una sequenza di **n** *prezzi giornalieri*, ciascuno relativo a una specifica giornata, e ha associato il numero di giorni in cui si è rilevato il prezzo giornaliero del titolo. Il prezzo giornaliero è composto da 4 valori:

- prezzo **open**, registrato la mattina, all'apertura della borsa (in euro)
- prezzo **high**, il prezzo massimo registrato in giornata (in euro)
- prezzo **low**, il prezzo minimo registrato in giornata (in euro)
- prezzo **close**, prezzo registrato la sera alla chiusura della borsa (in euro)

1. Si definiscono le strutture dati per rappresentare in linguaggio C un **prezzo giornaliero** e un **titolo societario**. Si ipotizzi che il numero massimo di prezzi giornalieri sia definito dalla costante MAX_PREZZI.
2. Si dichiari una variabile **mioTitolo** che rappresenti un titolo societario e le ulteriori variabili eventualmente necessarie, e si scriva una porzione di codice che la inizializzi richiedendo all'utente di inserire da tastiera il nome del titolo, il numero dei giorni da inserire e, per ciascun giorno, il valore dei quattro prezzi in euro di cui è composto. Un esempio di sessione al terminale è il seguente:

```
Inserire nome titolo e numero di giorni ===> AAPL 5
Inserire prezzo 'open high low close' del giorno n. 1: 167.99 167.90 162.97 163.49
Inserire prezzo 'open high low close' del giorno n. 2: 167.85 171.11 166.67 170.19
Inserire prezzo 'open high low close' del giorno n. 3: 166.78 167.87 165.59 167.55
Inserire prezzo 'open high low close' del giorno n. 4: 162.23 171.87 153.20 169.48
Inserire prezzo 'open high low close' del giorno n. 5: 167.24 167.27 161.80 163.01
```

3. Dichiarando le ulteriori variabili eventualmente necessarie, si scriva una porzione di codice che stampi il contenuto della variabile **mioTitolo**, aggiungendo un **asterisco** in fondo a ciascuna giornata nei casi in cui **close** > **open** per almeno **3** giorni consecutivi; nell'esempio precedente, la stampa a terminale dovrebbe quindi essere:

```
Analisi titolo AAPL (5 giorni):
open          high          low           close
----          ----          ----          ----
167.99        167.90        162.97        163.49
167.85        171.11        166.67        170.19
166.78        167.87        165.59        167.55
162.23        171.87        153.20        169.48 *
167.24        167.27        161.80        163.01
```

dove si nota che la quarta giornata ha un asterisco poiché, insieme alle due precedenti, il titolo in esame ha chiuso in rialzo rispetto all'apertura.

Soluzione

1)

```
#define MAX_STRING_CHARS 50
#define MAX_PREZZI100
typedef struct {
	double open;
	double high;
	double low;
	double close;
} prezzoGiornaliero;
typedef struct {
	char nomeTitolo[MAX_STRING_CHARS];
	prezzoGiornaliero prezzi[MAX_PREZZI];
	int numeroGiorni;
} titolo;
```

2)

```
titolo mioTitolo;
printf("Inserire nome titolo e numero di giorni ===> ");
scanf("%s %d", mioTitolo.nomeTitolo, &mioTitolo.numeroGiorni);
for (int i = 0; i < mioTitolo.numeroGiorni; i++) {
	printf("Inserire prezzo 'open high low close' del giorno n. %d: ", i+1);
	scanf("%lf %lf %lf %lf", &mioTitolo.prezzi[i].open, &mioTitolo.prezzi[i].high,
	&mioTitolo.prezzi[i].low, &mioTitolo.prezzi[i].close); }
```

3)

```
printf("\nAnalisi titolo %s (%d giorni): \n", mioTitolo.nomeTitolo,mioTitolo.numeroGiorni);
printf("%10s %10s %10s %10s\n", "open", "high", "low", "close");
printf("%10s %10s %10s %10s\n", "----", "----", "----", "----");
int conta = 0;
for (int i = 0; i < mioTitolo.numeroGiorni; i++) {
	printf("%lf %lf %lf %lf", mioTitolo.prezzi[i].open,
	mioTitolo.prezzi[i].high,mioTitolo.prezzi[i].low, mioTitolo.prezzi[i].close);
	if (mioTitolo.prezzi[i].close > mioTitolo.prezzi[i].open) {
		conta++;
		if (conta >= 3) {
			printf(" *\n");
		}
		else {
			printf(" \n");
		}
	}
	else {
		conta = 0;
		printf(" \n");
	}
}
```