

/*
* Corso di Fondamenti di Informatica
* Esercizio:
* elaborazione di polinomi rappresentati tramite array
*/

/*****

Scrivere un programma C che esegue la somma ed il prodotto tra due polinomi A e B nella variabile x. Il grado (si veda la Nota 0) massimo che A e B possono avere è definito da due parametri del programma, chiamati GRADO_MAX_A e GRADO_MAX_B, mentre il grado effettivo dei polinomi è inserito dall'utente. Per contenere i coefficienti dei polinomi utilizzare TRE array POLIA, POLIB e POLIRIS (quest'ultimo usato prima per la somma e poi per il prodotto) aventi dimensioni minime possibili.

Il programma deve:

1. gestire l'inserimento da tastiera del grado dei polinomi A e B;
2. gestire l'inserimento da tastiera dei coefficienti di A e B;
3. inserire la somma in POLIRIS e stampare il risultato (si veda la Nota 1);
4. inserire il prodotto in POLIRIS e stampare il risultato (si veda la Nota 1);

Nota 0: il grado di un monomio in una variabile e' l'esponente al quale la variabile e' elevata. Il grado di un polinomio e' il grado del suo monomio di grado massimo. Il termine noto di un polinomio corrisponde al suo monomio di grado 0.

Nota 1: ai punti 2 e 3 con l'espressione "stampare il risultato" si intende la stampa dei coefficienti dei termini del polinomio risultante (inclusi gli eventuali coefficienti nulli), in ordine di grado decrescente.

Nota 2: la somma di due polinomi ha grado pari al maggiore tra quelli dei polinomi addendi; il prodotto di due polinomi ha grado pari alla somma di quelli dei polinomi fattori.

Esempio di somma e prodotto tra polinomi con gradi 3 e 2:

$$\begin{array}{r} 3 \qquad 2 \\ (4.3x + x - 2.9x + 0.7) + (-x^2 + 1.3) = \\ 3 \qquad 2 \\ = 4.3x - 2.9x + 2.0 \end{array}$$

$$(4.3x^5 + x^4 - 2.9x^3 + 0.7x^2) (-x + 1.3) = -4.3x^5 - x^4 + 8.49x^3 + 0.6x^2 - 3.77x + 0.91$$

```
#include <stdio.h>

#define GRADO_MAX_A 10
/* massimo grado ammesso per il polinomio A */
#define GRADO_MAX_B 10
/* massimo grado ammesso per il polinomio B */
#define GRADO_MAX_RIS (GRADO_MAX_A+GRADO_MAX_B)
/* massimo grado che il polinomio risultato può assumere, dati i gradi dei due
 * polinomi addendi o fattori. Notare che il grado così definito è quello
 * del polinomio prodotto, ma è anche sicuramente maggiore o uguale al massimo
 * tra GRADO_MAX_A e GRADO_MAX_B: dunque è sicuramente almeno pari al grado del
 * polinomio somma. In definitiva, un array in grado di rappresentare un
 * polinomio di grado GRADO_MAX_RIS è adatto per contenere sia il prodotto che
 * la somma dei polinomi A e B */

int main()
{
    float Polia[GRADO_MAX_A+1];
    /* polinomio A: l'elemento dell'array di indice i rappresenta il
     * coefficiente del termine di grado i. Solo gli elementi dell'array aventi
     * indice compreso tra 0 e il grado del polinomio vengono effettivamente
     * utilizzati. */
    float Polib[GRADO_MAX_B+1];
    /* polinomio B: l'elemento dell'array di indice i rappresenta il
     * coefficiente del termine di grado i. Solo gli elementi dell'array aventi
     * indice compreso tra 0 e il grado del polinomio vengono effettivamente
     * utilizzati. */
    float Poliris[GRADO_MAX_RIS+1];
    /* polinomio risultato della somma o del prodotto: l'elemento dell'array di
     * indice i rappresenta il coefficiente del termine di grado i. Solo gli
     * elementi dell'array aventi indice compreso tra 0 e il grado del polinomio
     * vengono effettivamente utilizzati. */
    int Grada, Gradb, Gradoris;
    /* gradi effettivi dei polinomi A e B (inseriti dall'utente) e grado
```

```
* effettivo del polinomio risultante dal prodotto tra A e B */
int GradoCorra, GradoCorrB, GradoCorrRis;
/* contatori usati per scorrere gli array (il terzo non è necessario, ma
 * introdurlo chiarifica il programma) */
/* -----inizializzazione dei polinomi A e B:----- */
printf("\nInserisci il grado del polinomio A:\n");
scanf("%d", &GradoA);
printf("\nInserisci il grado del polinomio B:\n");
scanf("%d", &GradoB);
GradoRis = GradoA + GradoB;
/* per le regole del prodotto tra polinomi */
printf("\nInserisci i coefficienti del polinomio A:\n");
GradoCorra = GradoA;
while (GradoCorra >= 0)
{
    printf("\nA%d (coeff. di x^%d): ", GradoCorra, GradoCorra);
    scanf("%f", &Polia[GradoCorra]);
    --GradoCorra;
}

printf("\nInserisci i coefficienti del polinomio B:\n");
GradoCorrB = GradoB;
while (GradoCorrB >= 0)
{
    printf("\nB%d (coeff. di x^%d): ", GradoCorrB, GradoCorrB);
    scanf("%f", &Polib[GradoCorrB]);
    --GradoCorrB;
}

/* -----stampa dei coefficienti dei polinomi A e B:----- */
printf("\nA:\n");
GradoCorra = GradoRis;
while (GradoCorra >= 0)
{
    if (GradoA >= GradoCorra )
        /* se il polinomio possiede un termine di grado GradoCorra */
        printf("%.3f x^%d\t", Polia[GradoCorra], GradoCorra);
}
```

```

/* Nota: lo specificatore ".3" prima di "%f" serve a limitare a 3 il
 * numero di cifre dopo la virgola che vengono stampate; il segno + e'
 * un indicatore che richiede a printf di indicare sempre il segno del
 * numero stampato, anche quando e' positivo. */
else
}
{
printf("\t\t");
}
--GradoCorrA;
}

printf("\n\nB:\t");
GradoCorrB = GradORis;
while (GradoCorrB >=0)
{
if (GradoB >= GradoCorrB )
/* se il polinomio possiede un termine di grado GradoCorrB */
{
printf("%+.3f x^%d\t", PolIB[GradoCorrB], GradoCorrB);
}
else
{
printf("\t\t");
}
--GradoCorrB;
}
}

/* -----calcolo e stampa della somma:----- */
GradoCorrRis=0;
while (GradoCorrRis <= GradORis)
{
if ( (GradoCorrRis > GradoA) && (GradoCorrRis > GradoB) )
/* se nessuno dei due polinomi A e B include monomi di grado
 * GradoCorrRis */
{
PolIRis[GradoCorrRis] = 0.0;
}
else
/* almeno uno tra i polinomi A e B contiene un monomio di grado
 * GradoCorrRis */

```



```
    }
    --GradoCorrRis;
}

/* -----calcolo e stampa del prodotto:----- */
GradoCorrRis = 0;
while (GradoCorrRis <= GradORis)
{
    /* azzerramento di tutti i coefficienti del risultato */
    PolIRis[GradoCorrRis] = 0.0;
    ++GradoCorrRis;
}

GradoCorra = 0;
while (GradoCorra <= GradoA)
{
    GradoCorrB = 0;
    while (GradoCorrB <= GradoB)
    {
        PolIRis[GradoCorra+GradoCorrB] = PolIRis[GradoCorra+GradoCorrB] +
        PolIA[GradoCorra] * PolIB[GradoCorrB];
        ++GradoCorrB;
    }
    ++GradoCorra;
}

printf("\n\nA*B:\n");
GradoCorrRis = GradORis;
while (GradoCorrRis >=0)
{
    printf("%.3f x^%d\t", PolIRis[GradoCorrRis], GradoCorrRis);
    --GradoCorrRis;
}

printf("\n\n");
return(0);
}

/* ESERCIZIO A
* In questo programma, il codice relativo alla somma di polinomi e' piuttosto
```

- * complicato, il che rende piu' probabili gli errori di programmazione e piu' difficile il debugging. Si modifichi il programma in modo da semplificare tale codice.
- * [Suggerimento: il problema e' causato dal fatto che gli elementi non utilizzati degli array PolIA e PolIB vengono lasciati non inizializzati e dunque occorre ignorarli durante il calcolo.] */

/* ESERCIZIO B (da svolgere non appena a lezione sara' stato trattato l'argomento delle funzioni C)

- * Questo programma contiene parecchie operazioni ripetute, ovvero identiche nello svolgimento ma applicate a dati diversi. Ad esempio, la lettura dei coefficienti di un polinomio avviene due volte, una per il polinomio A e una per il polinomio B. Anche l'operazione di stampa di un polinomio viene eseguita più volte per polinomi diversi.
- * Modificare il programma per implementare ciascuna di queste operazioni ripetute tramite una opportuna FUNZIONE. */