```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * elaborazione di polinomi rappresentati tramite array
 * (Soluzione ESERCIZIO B)
 */

/*************************************************************************
 * Versione del programma rielaborata, che fa uso di funzioni. (Si veda
 * l'ESERCIZIO B in coda alla versione originale del programma.)
 *************************************************************************/

#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 */


/************** prototipi delle funzioni ***************/
int InputPolinomio(float PoliDaLeggere[]);
/* legge il grado di un polinomio e i valori dei suoi coefficienti; inserisce
 * questi ultimi nell'array, e restituisce il grado del polinomio letto */
void StampaPolinomio(float PoliDaStamp[], int GradoPoli, int GradoMax);
/* stampa a schermo il polinomio ricevuto in ingresso, avente grado GradoPoli,
 * opportunamente formattato per allineare i termini a quelli di un polinomio
 * di grado GradoMax */


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 */
int GradoA, GradoB, 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;
/* durante il calcolo del polinomio risultante da una somma o prodotto di
 * polinomi rappresentano i gradi dei termini correntemente in fase di
 * elaborazione, rispettivamente per i due polinomi operandi e per il
 * polinomio risultato */

/* ------------inizializzazione dei polinomi A e B:------------ */
printf("\n*** Inserimento del polinomio A ***");
GradoA = InputPolinomio(PoliA);

printf("\n*** Inserimento del polinomio B ***");
GradoB = InputPolinomio(PoliB);

GradoRis = GradoA + GradoB;
/* per le regole del prodotto tra polinomi */

/* ------------stampa dei coefficienti dei polinomi A e B:------------ */
printf("\n\nA:\t");
StampaPolinomio(PoliA, GradoA, GradoRis);

printf("\n\nB:\t");
StampaPolinomio(PoliB, GradoB, GradoRis);

/* ------------calcolo e stampa della somma:------------ */
GradoCorrRis=0;
while (GradoCorrRis <= GradoRis)
```

```
{
  if ( (GradoCorrRis > GradoA) && (GradoCorrRis > GradoB) )
      /* se non esistono monomi addendi di grado GradoCorrRis */
      {
        PoliRis[GradoCorrRis] = 0;
      }
  else
      {
    if ( GradoCorrRis > GradoA )
          /* se esistono monomi addendi solo nel polinomio B */
    {
      PoliRis[GradoCorrRis] = PoliB[GradoCorrRis];
    }
        else
    {
      if ( GradoCorrRis > GradoB )
              /* se esistono monomi addendi solo nel polinomio A */
              {
                PoliRis[GradoCorrRis] = PoliA[GradoCorrRis];
              }
      else
              /* se esistono in entrambi i polinomi */
              {
                PoliRis[GradoCorrRis] = PoliA[GradoCorrRis] +
                                PoliB[GradoCorrRis];
            }
        }
        }
        ++GradoCorrRis;
}

printf("\n\nA+B:\n");
StampaPolinomio(PoliRis, GradoRis, GradoRis);


/* ------------calcolo e stampa del prodotto:------------ */
GradoCorrRis = 0;
while (GradoCorrRis <= GradoRis)
{
  /* azzeramento di tutti i coefficienti del risultato */
  PoliRis[GradoCorrRis] = 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");
  StampaPolinomio(PoliRis, GradoRis, GradoRis);

  printf("\n\n");

  return(0);
}


/************** corpo delle funzioni ***************/

int InputPolinomio(float PoliDaLeggere[])
{
  int Grado;
  /* grado del polinomio */
  int GradoCorr;
  /* grado del termine attualmente in fase di inserimento */

  printf("\nInserisci il grado del polinomio:\n");
  scanf("%d", &Grado);

  printf("\nInserisci i coefficienti del polinomio:\n");
  GradoCorr = Grado;
  while (GradoCorr >= 0)
```

```
  {
    printf("\nCoefficiente di x^%d): ", GradoCorr);
    scanf("%f", &PoliDaLeggere[GradoCorr]);
    --GradoCorr;
  }

  return(Grado);
}


void StampaPolinomio(float PoliDaStamp[], int GradoPoli, int GradoMax)
{
  int GradoCorr;
  /* grado del termine attualmente in fase di stampa */
  GradoCorr = GradoMax;

  while (GradoCorr >=0)
  {
    if (GradoPoli >= GradoCorr )
        /* se il polinomio possiede un termine di grado GradoCorrA */
    {
          printf("%+.3f x^%d\t", PoliDaStamp[GradoCorr], GradoCorr);
      /* 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");
        }
        --GradoCorr;
  }
}


/* ESERCIZIO.
 * Questo programma produce esattamente lo stesso output del programma da cui
 * deriva, eccetto che nella stampa del polinomio somma vengono stampati (con
 * coefficienti nulli) anche termini di grado superiore ai gradi massimi dei
 * polinomi addendi. Modificare il programma in modo tale che, durante la stampa
```

```
* del polinomio somma dei polinomi A e B, i termini aventi grado maggiore sia
* del grado di A che del grado di B non vengano stampati. Le modifiche devono
* essere fatte esclusivamente nella funzione main, senza toccare le altre
* funzioni. */
```