

Il programma esegue la somma di **k frazioni**. Per memorizzare le frazioni si utilizzano due vettori di 10 elementi: l'**array n** per i numeratori, l'**array d** per i denominatori (che non devono essere nulli).

Le frazioni da sommare possono essere:

- proprie $n[i] < d[i]$ (3/5),
- improprie $n[i] > d[i]$ (7/4),
- apparenti $n[i]$ multiplo di $d[i]$ ($8/2 = 4$)

Il programma, semplificate le frazioni di partenza, le espone a video, calcola la frazione somma e la visualizza dopo eventuale riduzione ai minimi termini:

$$1/4 + 5/4 + 1/2 + 7/1 = 36/4 = 9$$

e/o ed eventuale semplificazione a somma di un intero e di una frazione propria:

$$1/3 + 2/3 + 4/3 = 7/3 = 2 + 1/3$$

I passi effettuati sono i seguenti:

- lettura di k e delle k frazioni negli array **n**[10] e **d**[10]
- semplificazione delle frazioni (numeratore e denomin. vanno divisi per il proprio M.C.D. se maggiore di 1)
- calcolo del denominatore della frazione somma (**den**): è il m.c.m. dei denominatori
- calcolo del numeratore (**num**) della frazione somma $den/d[0]*n[0]+den/d[1]*n[1]+ \dots +den/d[k-1]*n[k-1]$
- semplificazione della frazione risultato (num e den vanno divisi per il proprio M.C.D. se maggiore di 1)
- scrittura a video delle frazioni da sommare (input) e della frazione risultato
- se num > den, espressione del risultato come somma di un intero (risultato divisione intera tra num e den) e di una frazione propria ($num\%den$ fratto den)

representazione in **PSEUDOCODIFICA**

Funzione/programma principale (main)

```

INIZIO
  scriviTitolo()
  LEGGI k
  leggiFrazioni(n, d, k)
  scriviFrazioni(n, d, k)
  den ← d[0]
  PER i DA 1 A k-1 PASSO 1
    den ← mcm (den, d[i])
  num ← 0
  PER i DA 0 A k-1 PASSO 1
    num ← num + (den / d[i]) * n[i]
  scriviOperFraz (EQ, num, den)
  vfyScriviRid (num, den)
FINE
    
```

Funzione scriviTitolo()

NON restituisce valori, NON ha parametri in ingresso

```

INIZIO
  SCRIVI ("----- somma di frazioni ")
  . . .
FINE
    
```

Funzione leggiFrazioni (Rif num[], Rif den[], Val ele)

NON restituisce valori, ha 3 parametri in ingresso: 2 array di interi (passati per riferimento) di ele valori

```

INIZIO
  PER i DA 0 A ele-1 PASSO 1
    INIZIO
      LEGGI (num[i], den[i])
      semplFraz (num[i], den[i])
    FINE
  FINE
    
```

Funzione scriviFrazioni (Rif n[], Rif d[], Val k) come leggiFrazioni

```

INIZIO
  scriviFraz (n[0], d[0])
  PER i DA 1 A k-1 PASSO 1
    scriviOperFraz (PIU, n[i], d[i])
  FINE
    
```

Funzione scriviOperFraz (Val op, Val n, Val d)

NON restituisce valori, 3 parametri passati per valore il 1.parametro è di tipo carattere, 2. e 3. di tipo intero

```

INIZIO
  SCRIVI (SP, op, SP)
  scriviFraz (n, d)
FINE
    
```

Funzione scriviFraz (Val nu, Val de)

NON restituisce valori, ha 2 parametri(input) di tipo intero

```

INIZIO
  SCRIVI (nu)
  SE (de <> 1) ALLORA
    SCRIVI (DIV, de)
  FINE SE
FINE
    
```

Funzione semplFraz (Rif up, Rif dwn)

Restituisce il M.C.D. (maxcodi) di numer. e denom. quindi la funzione è di tipo intero i parametri sono passati per riferimento quindi la modifiche sui parametri formali (up, dwn) modificano i parametri attuali, cioè le variabili della funzione chiamante (leggiFrazioni e vfyScriviRid)

```

INIZIO
  maxcodi ← MCD (up, dwn)
  SE (maxcodi <> 1) ALLORA
    INIZIO
      up ← up / maxcodi
      dwn ← dwn / maxcodi
    FINE
  RITORNO-RESTITUISCI maxcodi
FINE
    
```

Funzione vfyScriviRid (Val num, Val den)

NON restituisce valori, ha 2 parametri(input) di tipo intero

Richiama la funzione *semplFraz* passando la frazione risultato (num, den); se questa viene semplificata (in tal caso il MCD restituito è diverso da 1) o se risulta una frazione impropria, la funzione provvede alla scrittura della frazione ridotta o semplificata in una somma di intero e di una frazione propria.

```

#include <iostream>
using namespace std;
#define PIU '+'
#define EQ '='
#define DIV '/'
#define SP ' '
#define DIM 10
int mcm (int alfa, int beta){
    int ma, mb;
    ma = alfa;
    mb = beta;
    while (ma != mb) {
        if (ma > mb)
            mb = mb + beta;
        else
            ma = ma + alfa;
    }
    return ma;
}
int MCD (int uno, int due){
    int num, dvs, res;
    num = uno;
    dvs = due;
    do {
        res = num % dvs;
        num = dvs;
        dvs = res;
    } while (dvs != 0);
    return num;
}
void scriviTitolo (void) {
    cout<< endl;
    cout<< "----- somma di frazioni ";
    cout<< "(minimo 2 massimo 10) -----";
    cout<< endl;
    cout<< "    proprie (3/5),";
    cout<< " improprie (7/4),";
    cout<< " apparenti (8/1)\n\n";
}
void scriviFraz (int nu, int de) {
    cout<< nu;
    if (de == 1) //--se denominatore vale 1
        return; //--- uscita anticipata dalla funzione
    cout<< DIV;
    cout<< de; //-- scrittura denominatore
}
void scriviOperFraz (char op, int n, int d) {
    cout<< SP << op << SP;
    scriviFraz (n, d);
}
int semplFraz (int& up, int& dwn) {
    int maxcodi;
    maxcodi = MCD (up, dwn);
    if (maxcodi != 1) {
        up = up / maxcodi;
        dwn = dwn / maxcodi;
    }
    return maxcodi;
}

```

```

void vfyScrivRid (int num, int den) {
    int resto=0, mcdiv;
    char operaz = EQ;
    mcdiv = semplFraz (num, den);
    if (mcdiv != 1 || num > den)
        { if (num > den && den != 1)
            {cout<< SP << operaz << SP;
              resto = num%den;
              num = num/den;
              cout << num;
              operaz = PIU;
              num = resto;
            }
        }
    scriviOperFraz (operaz, num, den);
}
void leggiFrazioni (int num[], int den[], int ele){
    for (int i=0; i<ele; i++) {
        cout<<"num. fraz."<< i << "= ";
        cin>>num[i];
        cout<<"den. fraz."<< i << "= ";
        do
            cin>>den[i]; //- zero NO ! ---
            while (den[i] == 0);
//--- semplificazione frazioni -----
        semplFraz (num[i], den[i]);
    }
}
void scriviFrazioni (int n[], int d[], int k){
    cout<< endl;
    scriviFraz (n[0], d[0]);
    for (int i=1; i<k; i++)
        scriviOperFraz (PIU, n[i], d[i]);
}
main(){
    int i, k, num, den;
    int n[DIM], d[DIM];
    scriviTitolo();
//--- lettura dati di input -----
    do {
        cout<<"quante frazioni? (2-10): ";
        cin>>k;
    } while (k < 2 || k > DIM);
    leggiFrazioni (n, d, k);
//--- scrittura frazioni di input ---
    scriviFrazioni (n, d, k);
//--- calcolo frazione risultato ----
    den = d[0];
    for (i=1; i<k; i++)
        den = mcm (den, d[i]);
    num = 0;
    for (i=0; i<k; i++)
        num = num + (den / d[i]) * n[i];
//--- scrittura frazione risultato --
    scriviOperFraz (EQ, num, den);
//--- verifica riduzione frazione ---
    vfyScrivRid (num, den);
    cout<< endl;
}

```