



Programmazione I

Il Linguaggio C

Esercizi

Daniel Riccio

Università di Napoli, Federico II

07 dicembre 2021



Sommario



- Esercizi

- Esercizi

Esercizio – Operazioni sulle frazioni



Realizzare un programma che permetta di eseguire **somme**, **sottrazioni**, **moltiplicazioni** e **divisioni** tra coppie di **frazioni** e di visualizzare il risultato in forma ridotta. Una frazione è una coppia di interi **n1** (**numeratore**), **d1** (**denominatore positivo**).

Scrivere le seguenti funzioni:

- **leggi_frazione** – Gestisce l'inserimento di una frazione da tastiera nella forma intero/intero positivo. Controlla che l'inserimento sia corretto, altrimenti chiede all'utente di reinserire i dati. Restituisce la frazione come coppia di interi.
- **leggi_operazione** – Gestisce l'inserimento da tastiera dell'operazione aritmetica da effettuare tra le due frazioni inserite. Legge il carattere inserito dall'utente, salta caratteri \n e permette il re-inserimento in caso di errore.
- **somma_frazioni** – Somma frazioni rappresentate da coppie di interi. La frazione somma è data in output attraverso i parametri della funzione.
- **moltiplica_frazioni** – Come sopra ma per il prodotto.
- **massimo_comun_divisore** – Restituisce il massimo comun divisore tra numeratore e denominatore di una frazione.
- **riduci_frazione** – Riduce una frazione dividendo il suo numeratore e denominatore per il loro massimo comun divisore.
- **stampa_frazione** – isualizza la frazione come numeratore/denominatore.
- **main** – gestisce un menu a scelta multipla per la selezione da parte dell'utente di effettuare un' operazione di somma '+', sottrazione '-', moltiplicazione '*' e divisione '/'.



Esercizio – Operazioni sulle frazioni



Scriviamo l'header file **frazioni.h**

```
#ifndef FRAZIONI_H
#define FRAZIONI_H

// Definizioni di tipi
typedef struct {
    int num;    //numeratore
    int den;    //denominatore
} frazione;

// Prototipi di funzioni
void leggi_frazione(frazione *f);
char leggi_operazione(void);
void somma_frazioni(frazione f1, frazione f2, frazione *f3);
void moltiplica_frazioni(frazione f1, frazione f2, frazione *f3);
int massimo_comun_divisore (frazione f);
void riduci_frazione(frazione *f);
void inverti_frazione(frazione *f);
void stampa_frazione(frazione f);

#endif
```

Esercizio – Operazioni sulle frazioni



```
int main()
{
    frazione f1, f2, f3;
    char op;
    char ripeti;

    printf("Questo programma prende in input:\n");
    printf("- due frazioni\n");
    printf("- una operazione aritmetica\n");
    printf("stampa in output la frazione risultante\n");

    do {

        <DA SCRIVERE>

        // Chiediamo se deve essere eseguito un altro ciclo
        printf("\nEsegui un'altra operazione? (y/n)> ");
        scanf(" %c", &ripeti);
    } while (toupper(ripeti) == 'Y');

    return (0);
}
```

Esercizio – Operazioni sulle frazioni

<DA SCRIVERE>

leggi la prima frazione;

leggi l'operazione;

leggi la seconda frazione;

esegui l'operazione;

riduci la frazione;

stampa la frazione risultante;



Esercizio – Operazioni sulle frazioni

```
int main()
{
    ...

    leggi_frazione(&f1);

    op = leggi_operazione();

    leggi_frazione(&f2);

    printf("L'operazione da eseguire è: %d/%d %c %d/%d", f1.num, f1.den,
op, f2.num, f2.den);
```

Esercizio – Operazioni sulle frazioni

```
int main()
{
    ...

    /* Calcola il risultato */
    switch (op) {
        case '+':
            somma_frazioni(f1, f2, &f3);
            break;
        case '-':
            f2.num = -f2.num;
            somma_frazioni(f1, f2, &f3);
            break;
        case '*':
            moltiplica_frazioni(f1, f2, &f3);
            break;
        case '/':
            inverti_frazione(&f2);
            moltiplica_frazioni(f1, f2, &f3);
    }

    riduci_frazione(&f3);
}
```


Esercizio – Operazioni sulle frazioni

```
int main()  
{
```

```
    ...
```

```
    // Mostra il risultato  
    printf("\n");  
    stampa_frazione(f1);  
    printf(" %c ", op);  
    stampa_frazione(f2);  
    printf(" = ");  
    stampa_frazione(f3);
```



Esercizio – leggi_frazione



```
/*
 * Legge da input una nuova frazione
 * Una frazione valida ha la forma: intero/intero positivo
 * Precondizioni: nessuna
 */
void leggi_frazione(frazione *f)
{
    char slash;    /* separatore fra numeratore e denominatore */
    int  stato;    /* variabili per il controllo dell'output della funzione scanf */
    int  errore;   /* flag che indica la presenza di un errore */
    char scarto;   /* carattere di input scartato */
```

Esercizio – leggi_frazione

```
void leggi_frazione(frazione *f)
```

```
{
```

```
    ...
```

```
do {
```

```
    <DA SCRIVERE>
```

```
    /* Scarta ulteriori caratteri in input */
```

```
do {
```

```
    scanf("%c", &scarto);
```

```
} while (scarto != '\n');
```

```
} while (errore);
```

Esercizio – leggi_frazione



<DA SCRIVERE>

```
/* nessun errore */
errore = 0;

/* leggi una frazione */
printf("Inserisci una frazione (due interi separati da uno slash): ");
stato = scanf("%d %c %d",&f->num, &slash, &f->den);

/* controlla il formato è valido */
if (stato < 3) {
    errore = 1;
    printf("Formato non valido segui le istruzioni\n");
} else if (slash != '/') {
    errore = 1;
    printf("Formato non valido i due interi devono essere separati da uno slash\n");
} else if ( f->den <= 0) {
    errore = 1;
    printf("Formato non valido, il denominatore deve essere un intero positivo\n");
}
```

Esercizio – leggi_operazione



```
/*
 * Legge l'operazione aritmetica da eseguire
 */
char leggi_operazione(void)
{
    char op;

    printf("Inserisci un operatore aritmetico (+,-,*, or /)\n> ");
    scanf("%c", &op);

    while(op != '+' && op != '-' && op != '*' && op != '/') {
        if (op != '\n')
            printf("%c non valido, reinserisci l'operatore (+,-, *,/)\n> ", op);

        scanf("%c", &op);
    }
    return (op);
}
```

Esercizio – Operazioni sulle frazioni



Alcune considerazioni riguardo alla somma di due frazioni

la **somma** di più frazioni aventi lo **STESSO DENOMINATORE** è una frazione che ha per numeratore la **SOMMA DEI NUMERATORI** e per denominatore lo **STESSO DENOMINATORE**.

la **somma** di due frazioni aventi **DIVERSO DENOMINATORE** si procede nel modo seguente:

- dapprima si **RIDUCE AL MINIMO COMUNE DENOMINATORE**;
- poi si procede come nel caso precedente.

Esercizio – somma_frazioni

```
void somma_frazioni(frazione f1, frazione f2, frazione *f3)
{
    int den;    // denominatore comune per la somma
    int num;    // numeratore della frazione risultante
    int segno;  // -1 negativo, 1 altrimenti

    // Troviamo il denominatore comune
    den = f1.den * f2.den;

    // Calcoliamo il numeratore
    num = f1.num * f2.den + f2.num * f1.den;

    // Impostiamo il segno corretto
    if (num * den >= 0)
        segno = 1;
    else
        segno = -1;

    num = segno * abs(num);
    den = abs(den);

    // Copiamo il risultato in f3
    f3->num = num;
    f3->den = den;
}
```

Esercizio – moltiplica_frazioni



```
void moltiplica_frazioni(frazione f1, frazione f2, frazione *f3)
{
    f3->num = f1.num * f2.num;
    f3->den = f1.den * f2.den;
}
```

```
void riduci_frazione(frazione *f3)
{
    int mcd;

    mcd = massimo_comun_divisore(*f3);
    f3->num = f3->num / mcd;
    f3->den = f3->den / mcd;
}
```


Esercizio – stampa_frazioni



```
void stampa_frazione(frazione f)
{
    printf("%d/%d", f.num, f.den);
}

void inverti_frazione(frazione *f)
{
    int temp;

    temp = f->num;
    f->num = f->den;
    f->den = temp;

    if(f->den < 0){
        f->den = abs(f->den);
        f->num = -(f->num);
    }
}
```



Esercizio – massimo_comun_divisore

```
int massimo_comun_divisore(frazione f)
{
    int r;
    int a;
    int b

    a = f.num;
    b = f.den;

    r = a % b;

    while(r>0){
        a = b;
        b = r;
        r = a % b;
    }

    return b;
}
```

Esempio: **a**=1705 e **b**=625

$$\begin{array}{r} 1705 : 625 = 2 \\ 455 \end{array}$$

$$\begin{array}{r} 625 : 455 = 1 \\ 170 \end{array}$$

$$\begin{array}{r} 455 : 170 = 2 \\ 115 \end{array}$$

$$\begin{array}{r} 170 : 115 = 1 \\ 55 \end{array}$$

$$\begin{array}{r} 115 : 55 = 2 \\ \boxed{5} \end{array}$$

$$\begin{array}{r} - - - - - \\ 55 : 5 = 11 \\ 0 \end{array}$$

Esercizio – esecuzione



```
Questo programma prende in input:
- due frazioni
- una operazione aritmetica
stampa in output la frazione risultante
Inserisci una frazione (due interi separati da uno slash): 1/5
Inserisci un operatore aritmetico (+,-,*, or /)
> +
Inserisci una frazione (due interi separati da uno slash): 2/15
L'operazione da eseguire è: 1/5 + 2/15
1/5 + 2/15 = 1/3
Esegui un'altra operazione? (y/n)> y
Inserisci una frazione (due interi separati da uno slash): 3/4
Inserisci un operatore aritmetico (+,-,*, or /)
> -
Inserisci una frazione (due interi separati da uno slash): 1/6
L'operazione da eseguire è: 3/4 - 1/6
3/4 - 1/6 = 7/12
Esegui un'altra operazione? (y/n)> y
Inserisci una frazione (due interi separati da uno slash): 5/4
Inserisci un operatore aritmetico (+,-,*, or /)
> *
Inserisci una frazione (due interi separati da uno slash): 4/15
L'operazione da eseguire è: 5/4 * 4/15
5/4 * 4/15 = 1/3
Esegui un'altra operazione? (y/n)> y
Inserisci una frazione (due interi separati da uno slash): 5/4
Inserisci un operatore aritmetico (+,-,*, or /)
> /
Inserisci una frazione (due interi separati da uno slash): 15/8
L'operazione da eseguire è: 5/4 / 15/8
5/4 / 15/8 = 2/3
Esegui un'altra operazione? (y/n)> n
```

Esercizio – Sequenza Corretta



Il programma seguente prende una stringa in input e verifica se essa contiene il carattere '**A**'

Esercizio – Sequenza Corretta



1	OK	<code>#include<stdio.h></code>
2		<code>#include<stdlib.h></code>
3	OK	<code>#define MAXSTR 80</code>
4		<code>#define CARATTERE 'A'</code>
6	OK	<code>int main()</code>
7	OK	<code>{</code>
8	OK	<code>char a[MAXSTR];</code>
9	OK	<code>int i, trovato=0;</code>
10		<code>float i, trovato=0;</code>
11	OK	<code>printf("Stringa? ");</code>
12	OK	<code>gets(a);</code>
13		<code>scanf("%s", &a);</code>
14	OK	<code>for (i=0; i<10 && a[i] != '\0'; i++)</code>
15		<code>while (i<10 && a[i] != '\0')</code>
16	OK	<code>if (a[i] == 'A')</code>
17	OK	<code>trovato = 1;</code>

18		<code>trovato = CARATTERE;</code>
19		<code>else</code>
20	OK	<code>if (trovato == 1)</code>
21	OK	<code>printf("Trovato\n");</code>
22	OK	<code>else</code>
23		<code>trovato = ' ';</code>
24	OK	<code>printf("Non trovato\n");</code>
25	OK	<code>return 0;</code>
26		<code>return;</code>
27	OK	<code>}</code>
28		<code>}</code>

Esercizio – Sequenza Corretta

Il programma seguente prende una stringa in input e verifica se essa contiene il carattere 'A'

```
#include<stdio.h>
#define MAXSTR 80

int main()
{
    char a[MAXSTR];
    int i, trovato=0;

    printf("Stringa? ");
    gets(a);
    for (i=0; i<10 && a[i] != '\0'; i++)
        if (a[i] == 'A')
            trovato = 1;
    if (trovato == 1)
        printf("Trovato\n");
    else
        printf("Non trovato\n");

    return 0;
}
```

Esercizio – Sequenza Corretta

Il programma seguente calcola la media di valori interi presi in input che siano compresi fra 18 e 30. Appena l'utente inserisce un valore fuori dall'intervallo, il programma stampa il risultato.



Esercizio – Sequenza Corretta



1	OK	#include <stdio.h>
2		#include <stdlib.h>
3	OK	int main()
4		{
5	OK	{
6	OK	int v, somma;
7		float v, somma;
8	OK	int cont;
9	OK	somma = 0;
10	OK	cont = 0;
11		while (v > 18 && v < 30);
12	OK	do
13	OK	{
14	OK	printf("Inserire il %do valore: ", cont+1);
15	OK	scanf("%d", &v);
16	OK	cont++;
16		--cont;
17	OK	somma += v;

18	OK	} while (v >= 18 && v <= 30);
19	OK	cont--;
20	OK	somma -= v;
21	OK	if (cont > 0)
22	OK	printf("Media: %f\n", (double)somma/cont);
23	OK	return 0;
24	OK	return;
25	OK	}
26		}



Esercizio – Sequenza Corretta

```
#include <stdio.h>
```

```
int main()
{
    int v, somma;
    int cont;

    somma = 0;
    cont = 0;
    do
    {
        printf("Inserire il %do valore: ", cont+1);
        scanf("%d", &v);
        cont++;
        somma += v;
    } while (v >= 18 && v <= 30);
    cont--;
    somma -= v;

    if (cont > 0)
        printf("Media: %f\n", (double)somma/cont);

    return 0;
}
```

Il programma seguente calcola la media di valori interi presi in input che siano compresi fra 18 e 30. Appena l'utente inserisce un valore fuori dall'intervallo, il programma stampa il risultato.

Esercizio – Esecuzione

```
#include <stdio.h>
```

```
int main()
{
    int v, max;
    int i, n;

    printf("Quanti? ");
    scanf("%d", &n);

    printf("Inserire il 1o valore: ");
    scanf("%d", &v);
    max = v;

    for (i=1; i<n; i++)
    {
        printf("Inserire il %do valore: ", i+1);
        scanf("%d", &v);
        if (v>max)
            max = v;
    }
    printf("Massimo: %d\n", max);

    return 0;
}
```

Qual è l'output del seguente programma quando l'input è:

5
23 43 12 76 34

Esercizio – Esecuzione



```
#include <stdio.h>
#include <stdlib.h>
#define MAXNUM 100
```

```
int main()
{
    int a[MAXNUM], t;
    int i, j, n;

    printf("Quanti valori? ");
    scanf("%d", &n);

    for (i=0; i<n; i++)
    {
        printf("inserisci il %do valore: ", i+1);
        scanf("%d", &a[i]);
    }
```

```
for (i=0, j=n-1; i<j; i++, j--)
{
    t = a[i];
    a[i] = a[j];
    a[j] = t;
}
```

```
for (i=0; i<n; i++)
    printf("%d ", a[i]);
printf("\n");
```

```
return EXIT_SUCCESS;
}
```

Qual è l'output del seguente programma quando l'input è:

```
5
23 43 12 76 34
```

Esercizio – Presenza di errori



```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define NUM 10;

double log2(double valore);

int main()
{
    int i;
    double a;

    for (i=0; i<NUM; i++)
    {
        printf("Introdurre un numero: ");
        scanf("%lf", a);
        printf("Il log2 di %f vale: %d\n", a, log2(a));
    }

    return a;
}
```

```
double log2(float x)
{
    float x;
    return log(x.0)/log(2.0);
}
```