

Titolo unità didattica: *Approccio incrementale*

[06]

Titolo modulo : *Function C per problemi aritmetici di base*

[06-C]

Sviluppo di function in C per il calcolo di somme e prodotti, ed esempi di utilizzo

Argomenti trattati:

- ✓ function in C per sommatorie
- ✓ function in C per sommatorie a segni alterni
- ✓ function in C per il fattoriale

Prerequisiti richiesti: *AP-05-03-C, AP-06-02-T, AP-06-05-AT*

problema:

calcolo della somma dei primi n numeri naturali

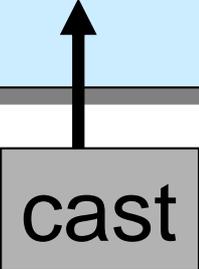
```
/* function per il calcolo della somma
   dei primi n numeri naturali          */
int somma_naturali(int n)
{
    int s, i;
    s = 0;
    for (i = 1; i <= n; i++)
        s = s + i;
    return s;
}
```

```
#include <stdio.h>
int somma_naturali(int n);
void main()
{
    int sn,n;
    printf("inserire il valore di n: ");
    scanf("%d",&n);
    sn = somma_naturali(n);
    printf("somma dei primi %d naturali: %d\n",n,sn);
    /* verifica via formula di Gauss */
    printf("verifica via Gauss: %d\n",n*(n+1)/2);
}
/* function somma dei primi n numeri naturali */
int somma_naturali(int n)
{
    int s,i;
    s = 0;
    for (i = 1; i <= n; i++)
        s = s + i;
    return s;
}
```

problema:

calcolo della somma dei reciproci dei primi n
numeri naturali

```
/* function somma dei reciproci dei primi n
   numeri naturali */
float somma_armonica(int n)
{
    float s;
    int i;
    s = 0.0F;
    for (i = 1; i <= n; i++)
        s = s + 1.0F / (float)i;
    return s;
}
```



cast

```
#include <stdio.h>
float somma_armonica(int n) ;
void main()
{
    int n;
    printf("inserire il valore di n: ");
    scanf("%d", &n) ;
    printf("somma armonica di %d termini:%f\n",
           n, somma_armonica(n)) ;
}
/* function somma dei reciproci dei primi n
   numeri naturali */
float somma_armonica(int n)
{
    float s;
    int i;
    s = 0.0F;
    for (i = 1; i <= n; i++)
        s = s + 1.0F/(float)i;
    return s;
}
```

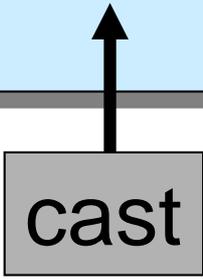
problema:

calcolo della somma dei quadrati dei reciproci dei primi n numeri naturali

$$v = \sum_{i=1}^n \frac{1}{i^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2}$$

```
/* function somma dei quadrati dei reciproci dei
   primi n numeri naturali */
float somma_quadr_rec(int n)
{
    float s;
    int i;
    s = 0.0F;
    for (i = 1; i <= n; i++)
        s = s + 1.0F / (float) (i*i);
    return s;
}
```

cast



```

#include <stdio.h>
#include <math.h>
float somma_quadr_rec(int n) ;
void main()
{
    int n;
    printf("inserire il valore di n: ");
    scanf("%d", &n) ;
    printf("somma quadr. rec. di %d termini:%f\n",
           n, somma_quadr_rec(n)) ;
}
/* function somma dei quadrati dei reciproci dei
   primi n numeri naturali */
float somma_quadr_rec(int n)
{
    float s;
    int i;
    s = 0.0F;
    for (i = 1; i <= n; i++)
        s = s + 1.0F/pow(i,2) ;
    return s;
}

```

problema:

calcolo della somma di un insieme di dati di input (immessi da tastiera)

```
#include <stdio.h>
void main()
{
    int n,i;
    float numero,s=0.0F;
    printf("inserire il numero dei numeri: ");
    scanf("%d",&n);
    for (i = 1; i <= n; i++)
    {
        printf("inserire %d-simo numero: ",i);
        scanf("%f",&numero);
        s = s + numero;
    }
    printf("somma dell'insieme di %d termini:%f\n",n,s);
}
```

problema:

calcolo della sommatoria a segni alterni

$$z = \sum_{i=1}^n \frac{(-1)^i}{2i+1}$$

```
/* function sommatoria a segni alterni */
float somma_segni_alt(int n)
{
    float s=0.0F, addendo, segno=-1.0F;
    int i;
    for (i = 1; i <= n; i++)
    {
        addendo = 1.0F/(2.0F*(float)i+1.0f);
        s = s + segno*addendo;
        segno = -segno;
    }
    return s;
}
```

conversioni automatiche in espressioni aritmetiche

```
int i;  
float f;
```

```
i + f
```

espressione
mista

espressioni
miste

```
i * f
```

```
i / f
```

```
i = f
```

```
f = i
```

il valore dell'espressione

```
i + f
```

è di tipo **float**

equivalente a

```
(float)i + f
```

il valore dell'espressione è di tipo **float**

il valore di **f** trasformato in **int** viene assegnato a **i**

il valore di **i** trasformato in **float** viene assegnato a **f**

conversioni automatiche in espressioni aritmetiche

```
int i, j;  
double d;
```

$i + d$

ha valore di tipo **double**

`(double) i+d`

```
float f;  
double d;
```

$f + d$

ha valore di tipo **double**

`(double) f+d`

espressioni
miste

$i * d$

i / d

$f * d$

f / d

hanno valore di tipo **double**

$f = i/j$

$d = i/j$

Attenzione: divisione intera in **double**

$i = d$

il valore di **d** trasformato in **int** viene assegnato a **i**

$d = i$

il valore di **i** trasformato in **double** viene assegnato a **d**

$f = d$

il valore di **d** trasformato in **float** viene assegnato a **f**

$d = f$

il valore di **f** trasformato in **double** viene assegnato a **d**

problema:
calcolo del fattoriale di un numero naturale

$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

$0! = 1$, $1! = 1$

```
int fattoriale (int n) {  
    int fatt ;  
    fatt = 1 ;  
    if (n>1) {  
        for (i=2, i<=n, i++)  
            { fatt = fatt * i ; }  
    }  
    return fatt ;  
}
```

```
int fattorialeI (int n)
```

```
float fattorialeF (int n)
```

scrivere un `main` che visualizza
il fattoriale di `n` calcolato
sia con `fattorialeI`
sia con `fattorialeF`
per `n` che varia da 1 a 14

```
/* function per il fattoriale di n (restituisce un int)*/  
int fattorialeI(int n)  
{  
    int fatt,i;  
    fatt = 1;  
    if (n > 1)  
        for (i = 2; i <= n; i++)  
            fatt = fatt * i;  
    return fatt;  
}
```

```
/*function per il fattoriale di n(restituisce un float)*/  
float fattorialeF(int n)  
{  
    int i;  
    float fatt = 1.0F;  
    if (n > 1)  
        for (i = 2; i <= n; i++)  
            fatt = fatt * (float)i;  
    return fatt;  
}
```

scrivere un `main` che visualizza il fattoriale di `n` calcolato sia con `fattorialeI` sia con `fattorialeF` per `n` che varia da 1 a 14

```
#include <stdio.h>
int fattorialeI(int);
float fattorialeF(int);
void main()
{
    int n, f_int;
    float f_float;
    printf("\n valore di n      fattorialeI      fattorialeF\n ");
    for (n = 1; n <= 14; n++)
    {
        f_int = fattorialeI(n);
        f_float = fattorialeF(n);
        printf("%7d %15d %15.0f \n",n,f_int,f_float);
    }
}
```

valore di n	fattorialeI	fattorialeF
1	1	1
2	2	2
3	6	6
4	24	24
5	120	120
6	720	720
7	5040	5040
8	40320	40320
9	362880	362880
10	3628800	3628800
11	39916800	39916800
12	479001600	479001600
13	1932053504	6227020800
14	1278945280	87178289152