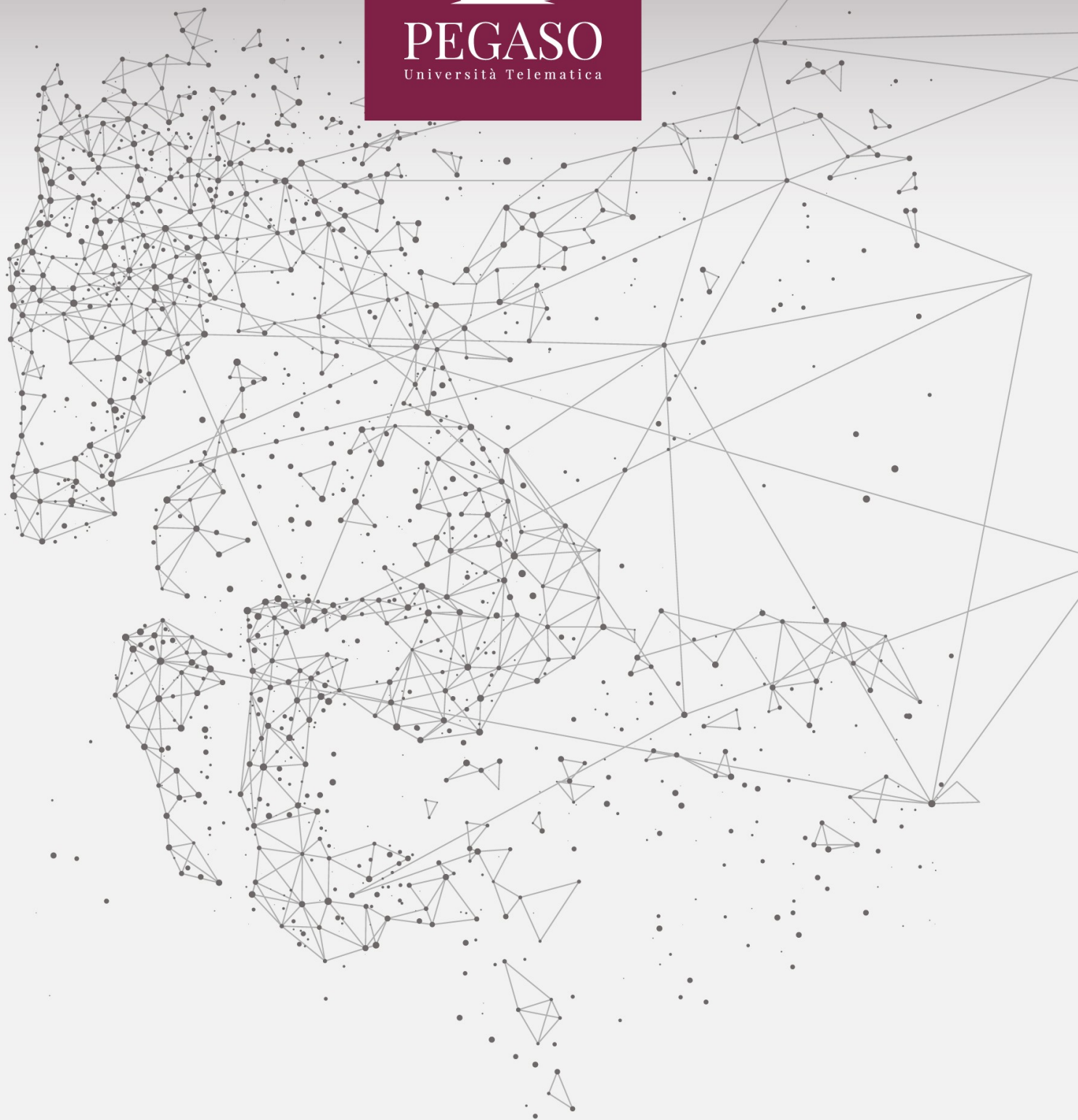




PEGASO
Università Telematica



«Il primo programma in C» Modestino Matarazzo

Il programma



Il programma

Si vuole scrivere un programma in linguaggio C che chieda all'utente di introdurre da tastiera due numeri interi e visualizzi il valore della loro somma sul video.

Il programma

Le operazioni da svolgere saranno le seguenti, nell'ordine indicato (quindi costituiscono l'algoritmo di soluzione di questo problema):

1. Chiedere il primo numero
2. Chiedere il secondo numero
3. Calcolare la somma del primo numero e del secondo numero
4. Visualizzare il risultato

Il programma

Le variabili

- I numeri richiesti vengono immagazzinati nella memoria centrale del calcolatore (RAM)
- Ciascun numero sarà immagazzinato in una piccola porzione di memoria detta **variabile** costituita da un certo numero di byte
- A ogni variabile il programmatore dà un nome per potersi riferire ad essa

Il programma

Sequenza delle operazioni da svolgere:

1. Chiedere il primo numero e metterlo in una variabile di nome A
2. Chiedere il secondo numero e metterlo in una variabile di nome B
3. Calcolare la somma di A e B e mettere il risultato in una variabile di nome S
4. Visualizzare il contenuto della variabile S

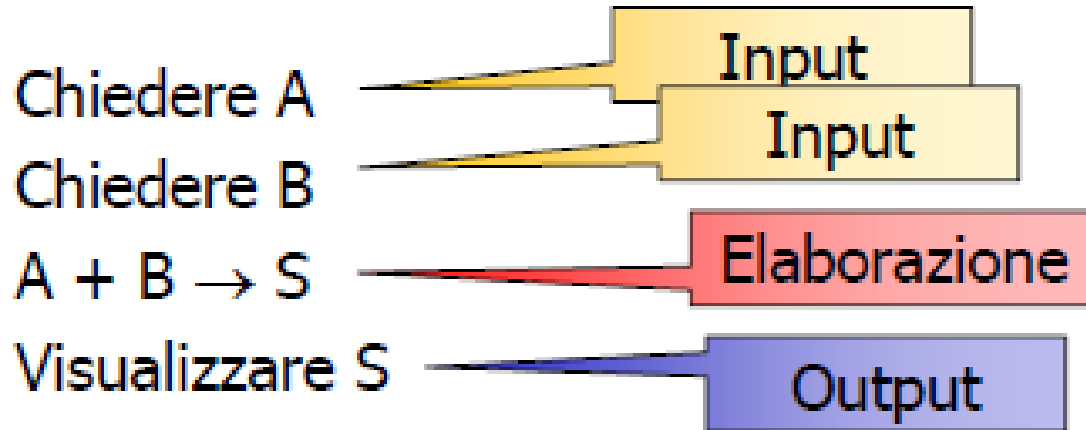
Il programma

Le variabili

Anche il risultato della somma dovrà essere immagazzinato nella memoria del calcolatore in una variabile, quindi avremo:

1. Chiedere il primo numero e metterlo in una variabile di nome A
2. Chiedere il secondo numero e metterlo in una variabile di nome B
3. Calcolare la somma di A e B e mettere il risultato in una variabile di nome S
4. Visualizzare il contenuto della variabile S

Il programma



Bozza di soluzione

- Input significa “mettere dentro”
- Le operazioni di input sono quelle tramite le quali il calcolatore riceve dei dati dall'esterno
- Per leggere un numero dalla tastiera e memorizzarlo in una variabile si può ad esempio utilizzare la funzione scanf:

scanf("%d", &A)

- Notare che la variabile A è preceduta dal carattere &
- La funzione precedente assegna ad A il valore **intero** letto dalla tastiera

Bozza di soluzione

- Output significa “mettere fuori”
- Le operazioni di output sono quelle tramite le quali il calcolatore mostra i risultati
- Per visualizzare il valore di una variabile si può utilizzare ad esempio la funzione printf:

printf("%d", S)

- Notare che la variabile S non è preceduta dal carattere &
- La funzione precedente visualizza un numero **intero**: quello contenuto nella variabile S

Bozza di soluzione

- Assegnare un valore ad una variabile significa memorizzare in essa un valore
- Per assegnare un valore si usa il simbolo '=':

$$\mathbf{X = 12}$$

- Il valore può essere il risultato di un calcolo:

$$\mathbf{S = A+B}$$

questo indica che in S viene memorizzato il valore risultante dal calcolo $A+B$

- Il segno = significa “prende il valore di”:
 S prende il valore del risultato di $A+B$
- Ogni volta che si assegna un valore ad una variabile, il contenuto precedente viene perso

Bozza di soluzione

```
scanf ("%d", &A)  
scanf ("%d", &B)  
S = A + B  
printf ("%d", S)
```

Bozza di soluzione

Le istruzioni devono terminare
sempre con un punto e virgola:

```
scanf ("%d", &A) ;
```

```
scanf ("%d", &B) ;
```

```
S = A + B ;
```

```
printf ("%d", S) ;
```

Bozza di soluzione

- Tutte le variabili utilizzate nel programma devono essere definite
- Le definizioni vanno collocate tutte insieme prima delle istruzioni eseguibili:

```
int A, B, S;
```

```
scanf("%d", &A);
```

```
scanf("%d", &B);
```

```
S = A + B;
```

```
printf("%d", S);
```

- int indica che A, B ed S sono variabili adeguate a contenere valori **interi**

Programma principale

```
main()  
{  
    int A, B, S;  
    scanf("%d", &A);  
    scanf("%d", &B);  
    S = A + B;  
    printf("%d", S);  
    return 0;  
}
```

Programma principale

Per far terminare il programma, si usa l'istruzione `return` (seguita da un numero, in genere 0) collocata alla fine del blocco `main`:

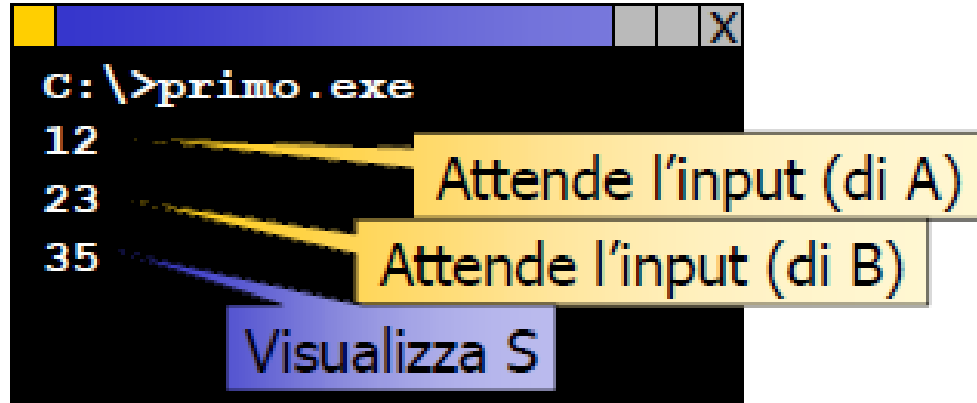
```
main()  
{  
    int A, B, S;  
    scanf("%d", &A);  
    scanf("%d", &B);  
    S = A + B;  
    printf("%d", S);  
    return 0;  
}
```

Programma principale

```
#include <stdio.h>

int main()
{
    int A, B, S;
    scanf("%d", &A);
    scanf("%d", &B);
    S = A + B;
    printf("%d", S);
    return 0;
}
```

Esecuzione



```
C: \>primo.exe
12
23
35
```

Attende l'input (di A)

Attende l'input (di B)

Visualizza S

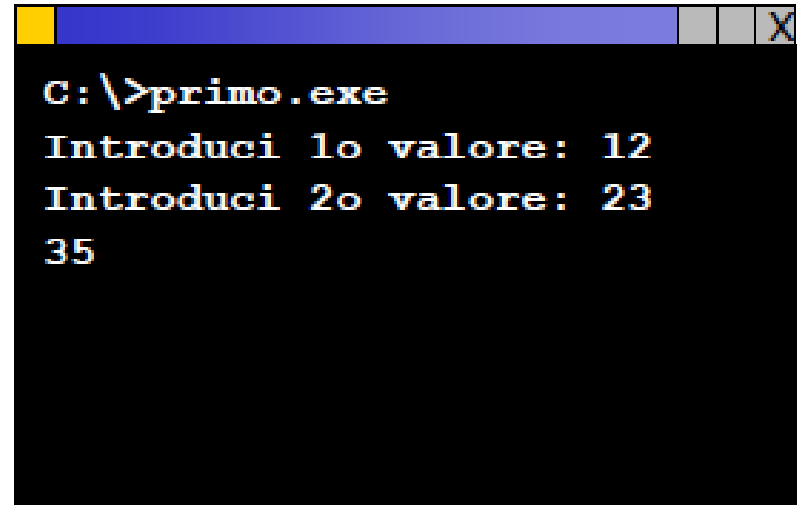
Esecuzione

- Quando il programma parte non informa l'utente di essere in attesa che egli introduca due valori da sommare
- È utile far precedere ciascuna `scanf` da un'istruzione `printf` che informi l'utente su che cosa egli debba fare:

```
printf("Introduci 1° valore: ");  
scanf("%d", &A);  
printf("Introduci 2° valore: ");  
scanf("%d", &B);
```

Esecuzione

- Dopo i miglioramenti indicati, all'esecuzione si ottiene la seguente schermata:



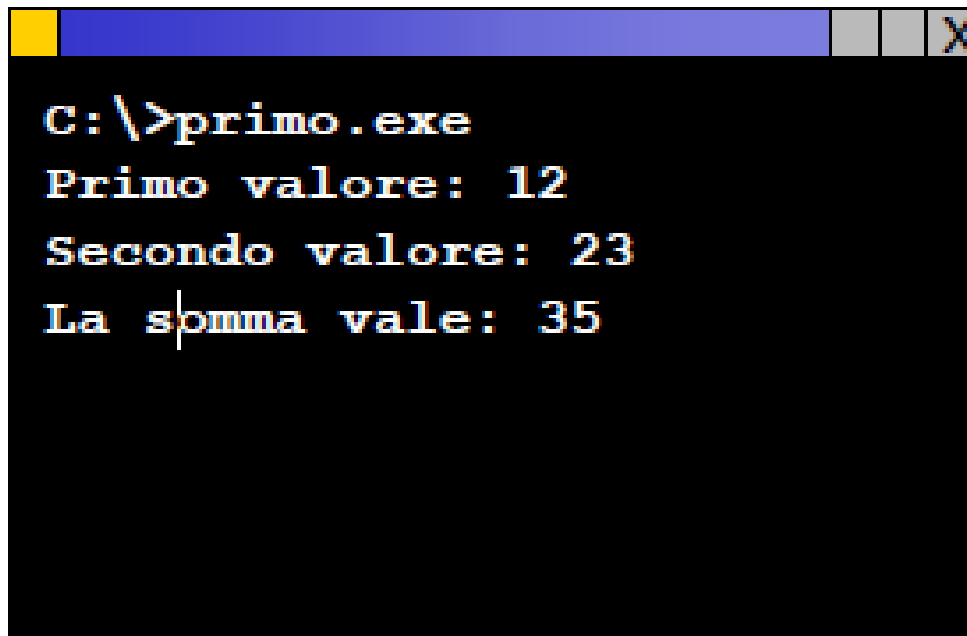
```
C:\>primo.exe
Introduci 1o valore: 12
Introduci 2o valore: 23
35
```

Esecuzione

- Il risultato del calcolo viene mostrato a video come semplice numero, ma è preferibile aggiungere una scritta che ne specifichi il significato
- Si può aggiungere tale scritta tra le virgolette della stessa printf che visualizza S:

```
printf("La somma vale: %d", S);
```

Esecuzione



```
C:\>primo.exe  
Primo valore: 12  
Secondo valore: 23  
La somma vale: 35
```

Programma completo

```
#include <stdio.h>
int main()
{
    int A, B, S;
    printf("Primo valore? ");
    scanf("%d", &A);
    printf("Secondo valore? ");
    scanf("%d", &B);
    S = A + B;
    printf("La somma vale: %d", S);
    return 0;
}
```