

Fondamenti di Informatica e Laboratorio T-AB e Fondamenti di Informatica T1
Ingegneria Elettronica e Telecomunicazioni e
Ingegneria dell'Automazione

Lab 03

Istruzioni

Prima Parte

Esercizio 1

- Realizzare un programma che legga da input tre numeri interi e stampi a video la loro somma e la media.

Esercizio 1 - Soluzione

```
#include <stdio.h>

int main()
{
    int num1, num2, num3, somma;
    float media;

    scanf("%d%d%d", &num1, &num2, &num3);

    somma = num1 + num2 + num3;
    media = somma / 3.0F;
    printf("Somma:%d\n", somma);
    printf("Media:%f\n", media);
    return 0;
}
```

scanf: STRINGA DI FORMATO

```
#include <stdio.h>
main()
{
    int intero1, intero2;
    float reale1;
    char car1, car2;

    scanf ("%d%d", &intero1, &intero2);
    printf ("%d,%d", intero1, intero2);
}
```

scanf: STRINGA DI FORMATO

```
scanf ("%d%d", &intero1, &intero2) ;
```

Inserire due interi separati da uno o più spazi:

12 35

12 35

scanf: STRINGA DI FORMATO

```
scanf ("%d,%d", &intero1, &intero2);
```

Inserire due interi separati da una (e una sola) virgola (eventuali spazi sono scartati):

12,35

12, 35

12 35

scanf: STRINGA DI FORMATO

Regole:

la stringa di formato descrive esattamente quello che deve esserci in input

lo spazio bianco viene considerato un separatore e viene scartato

Però...

Lo spazio bianco e' a tutti gli effetti un carattere...
quindi nella lettura di caratteri

```
scanf ("%d%c%c", &intero1, &car1, &car2) ;  
printf ("%d, %c, %c", intero1, car1, car2) ;
```

```
12 A B  
12, ,A
```

La scanf ha preso lo spazio come se fosse il
carattere inserito !

Una soluzione

Usare un separatore (anche lo spazio stesso)

spazio

`scanf("%d %c %c", &intero1, &car1, &car2);
printf("%d,%c,%c", intero1, car1,car2);`

```
12 A B  
12,A,B
```

Un'altra trappola

```
printf("Inserire un numero reale: ");
scanf("%f", &reale1);
printf("\nInserire un carattere: ");
scanf("%c", &car1);
printf("\nLetti: %f, %c", reale1, car1);
```

Questo frammento di programma sembra corretto...

Un'altra trappola

...ma il risultato e' questo:

Inserire un numero reale:
12.4

Inserire un carattere:
Letti: 12.400000,

Motivo

L' I/O e' bufferizzato: i caratteri letti da tastiera sono memorizzati in un buffer.

In architetture Windows, il tasto di INVIO corrisponde a 2 (DUE!) caratteri (CR LF): il primo è interpretato come separatore, ma il secondo rimane nel buffer ed è preso come carattere inserito dall'utente.

Una soluzione

Leggere il carattere "spurio"

```
printf("Inserire un numero reale: ");
scanf("%f", &reale1);
scanf("%*c"); /* letto e buttato via */
printf("\nInserire un carattere: ");
scanf("%c", &car1);
printf("\nLetti: %f,%c", reale1, car1);
```

Una Seconda Soluzione

Vuotare il buffer:

```
printf("Inserire un numero reale: ");
scanf("%f", &reale1);
fflush(stdin); /* Si vuota il buffer */
printf("\nInserire un carattere: ");
scanf("%c", &car1);
printf("\nLetti: %f,%c", reale1, car1);
```

Precisazione

Questo problema si verifica solo con la lettura di caratteri.

Negli altri casi il doppio carattere nel buffer e' considerato come sequenza di separatori e scartato.

Esercizio 2

- Realizzare un programma che legga da input tre caratteri consecutivi e ne stampi il valore a video (echo)

Esercizio 2 - Soluzione

```
#include <stdio.h>
int main()
{
    char c1, c2, c3;

    scanf ("%c%c%c", &c1, &c2, &c3);
    printf ("Caratteri:%c %c %c\n", c1, c2, c3);
    return 0;
}
```

Valutazione in cortocircuito (1)

- In C, le espressioni booleane sono valutate in cortocircuito (shortcut evaluation)
 - Appena è possibile determinare il valore logico dell' espressione, si salta la valutazione degli altri operandi
- Esempio:
 - lettura ciclica di due numeri reali
 - si esce dal ciclo quando è impossibile effettuare la divisione di essi, oppure il risultato della divisione è negativo

Valutazione in cortocircuito (2)

```
float a, b;  
do  
{  
    printf("Inserisci due numeri reali:");  
    scanf("%f %f", &a, &b);  
}  
while(b != 0 && a/b < 0);
```

- Senza la valutazione in cortocircuito, in caso di divisore==0 si effettuerebbe comunque la divisione (errore concettuale)

Esercizio 3

Scrivere, compilare ed eseguire il seguente programma:

```
int main() {  
    int a=5, b=5, c=5;  
    if (a>0 || a=a+1)  
        printf("%d", a);  
    if (b>0 && b=b+1)  
        printf("%d", b);  
    if (c>0 && c=c-5)  
        printf("%d", c);  
}
```

- Cosa viene stampato a video? Quanto valgono le variabili?

Esercizio 4 - switch

Stampa di voti

- Realizzare un programma che legge da input un voto (carattere tra ‘A’ ed ‘E’) e ne stampa il significato

Esercizio 4 – switch - soluzione

```
#include <stdio.h>
int main ()
{
    char voto;
    scanf("%c", &voto);
    switch(voto)
    {
        case 'A':
            printf("Ottimo\n"); break;
        case 'B':
            printf("Buono\n"); break;
        case 'C':
            printf("Sufficiente\n"); break;
        case 'D':
            printf("Insufficiente\n"); break;
        case 'E':
            printf("Gravemente insufficiente\n"); break;
        default:
            printf("Quale codifica hai usato?\n");
    }
    return 0;
}
```

Esercizio 5

- Realizzare un programma che legga da input un carattere dell'alfabeto e stampi a video il carattere stesso ed il suo valore ASCII
- Il programma deve controllare che il carattere inserito sia compreso tra ‘a’ e ‘z’ o tra ‘A’ e ‘Z’ (in caso contrario si stampi un messaggio di errore)
- Dopo la stampa, il programma deve continuare a chiedere nuovi caratteri, finché l’utente non inserisce il carattere corrispondente al numero zero (‘0’): in tal caso il programma termina.

Esercizio 5 - Soluzione

```
#include <stdio.h>
int main()
{
    char c1;

    do {
        printf("Inserisci un carattere alfabetico: ");
        scanf("%c%c", &c1);
        if ((c1>='a' && c1<='z') ||
            (c1>='A' && c1<='Z'))
            printf("%c %d\n", c1, c1);
        else
            if (c1 != '0')
                printf("Errore!\n");
    } while (c1!= '0');
    return 0;
}
```

Esercizio 6 – Cicli

Si realizzi un programma che, partendo da una base **a** ed un limite **n**, calcoli la seguente funzione:

$$\sum_{i=0}^n a^i$$

Realizzare il programma in due modi diversi:

- 1.Utilizzando due cicli (uno per la sommatoria, ed uno per la potenza)
- 2.Utilizzando un ciclo solo...

Esercizio 6 – Cicli - Soluzione

```
#include <stdio.h>

int main()
{
    int a, n, i, j;
    int somma;
    int prod;

    somma = 0; //elemento neutro della somma
    printf("Inserisci la base ed il numero di cicli: ");
    scanf("%d%d", &a, &n);

    for (i=0; i <= n; i++) {
        prod = 1; //elemento neutro del prodotto
        for (j=1; j <= i; j++) {
            prod = prod * a;
        }
        somma = somma + prod;
    }

    return 0;
}
```

Esercizio 6 – Cicli – Soluzione(variante)

```
#include <stdio.h>

int main()
{
    int a, n, i;
    int somma;
    int prod;

    somma = 0; //elemento neutro della somma
    prod = 1; //elemento neutro del prodotto
    printf("Inserisci la base ed il numero di cicli: ");
    scanf("%d%d", &a, &n);

    for (i=0; i <= n; i++) {
        if (i>0) {
            prod = prod * a;
        }
        somma = somma + prod;
    }

    return 0;
}
```

Esercizio 7

Si realizzi un programma che legga un intero N da tastiera, e stampi a video il risultato della seguente sommatoria:

$$\sum_{i=0}^N \left[(-1)^i \frac{4}{2*i + 1} \right]$$

Una volta calcolato e stampato il valore a video, il programma deve chiedere un nuovo numero all'utente e ripetere il calcolo. Il programma deve terminare solo qualora l'utente inserisca un valore negativo.

Esercizio 7

```
#include <stdio.h>

int main()
{
    int num, i, pari, potenza;
    double pi;

    do {
        pi = 0;
        printf("Inserire numero: ");
        scanf("%d", &num);
        if (num >=0) {
            for (i=0; i<=num; i++) {
                pari = i%2;
                if (pari) potenza = -1;
                else potenza = 1;
                pi = pi + potenza*(4.0/(2*i +1));
            }
            printf("Pi Greco vale: %f\n\n", pi);
        }
    } while (num >= 0);
    return 0; }
```

Esercizio 8

Stabilire il valore assoluto e la parte intera di un numero reale

- Realizzare un programma che legga da input un numero reale, e stampi a video:
 1. Il valore assoluto
 2. Il valore assoluto della sua parte intera

Esercizio 8 - Soluzione

```
#include <stdio.h>

int main()
{
    float num1, abs_real;
    int abs_int;

    scanf("%f", &num1);

    if (num1 > 0) abs_real = num1;
    else abs_real = - num1;

    abs_int = (int) abs_real;

    printf("Absolute value: %f\n", abs_real);
    printf("Absolute integer value: %d\n", abs_int);
    return 0;
}
```

Esercizio 9 - Cicli

Dato un numero intero **a**, definito tramite una variabile,
scrivere un programma che calcoli il valore della seguente
espressione:

$$\sum_{i=1}^a \sum_{j=1}^i j$$

Esercizio 9 - Cicli

```
#include <stdio.h>

int main()
{
    int a, i, j;
    int somma;

    printf("Inserisci un numero: ");
    scanf("%d", &a);

    somma = 0; //elemento neutro della somma
    for (i=1; i <= a; i++) {
        for (j=1; j <= i; j++) {
            somma = somma+j;
        }
    }

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

    return 0;
}
```

Esercizio 10 – if innestati

Stampa di caratteri in ordine alfabetico

- Realizzare un programma che legge da input tre caratteri, e li stampa in ordine alfabetico
- Utilizzando l'istruzione if
 - Per determinare il secondo carattere, devo per forza utilizzare degli if innestati

Esercizio 10 – if innestati - soluzione

```
#include <stdio.h>
int main ()
{
    char c1, c2, c3, first, second, third;
    int temp;
    scanf("%c%c%c", &c1, &c2, &c3);
    if(c1 <= c2)
    {
        first = c1;
        second = c2;
    }
    else
    {
        first = c2;
        second = c1;
    }
    ...
}
```

Esercizio 10 – if innestati - soluzione

```
...
if(c3 < first)
{
    third = second;
    second = first;
    first = c3;
}
else
{
    if(c3 < second)
    {
        third = second;
        second = c3;
    }
    else
    {
        third = c3;
    }
}
printf("Characters: %c %c %c\n", first, second, third);
return 0;
}
```

Esercizio 11 – if innestati

Triangoli

- Si scriva un programma, in linguaggio C, che legga da input le lunghezze dei lati di un triangolo e determini se il triangolo è equilatero, isoscele o scaleno

Esercizio 11 – if innestati - soluzione

```
#include <stdio.h>
int main() {
    double primo, secondo, terzo;
    printf("Lunghezze lati triangolo? ");
    scanf("%g%g%g", &primo, &secondo, &terzo);
    if (primo == secondo)
    {
        if (secondo == terzo) printf("Equilatero\n");
        else printf("Isoscele\n");
    }
    else
    {
        if (secondo == terzo)
            printf("Isoscele\n");
        else if (primo == terzo) printf("Isoscele\n");
        else printf("Scaleno\n");
    }
    return 0;
}
```

Esercizio 12 - switch

Operazioni aritmetiche

- Realizzare un programma che, presi in input 2 operandi reali e un operatore (+, -, *, /), esegue l' operazione stampandone il risultato
- Nel caso in cui l' operatore sia errato o l' operazione non eseguibile, si stampi “undefined”

Esercizio 12 - switch

```
#include <stdio.h>
int main ()
{
    float op1, op2;
    char type;
    scanf("%f %c %f", &op1, &type, &op2);
    switch(type)
    {
        case '+':
            printf("= %f\n", op1 + op2); break;
        case '-':
            printf("= %f\n", op1 - op2); break;
        case '*':
            printf("= %f\n", op1 * op2); break;
        case '/':
            op2 ? printf("= %f\n", op1 / op2) : printf("= undefined\n");
            break;
        default:
            printf("= undefined\n");
    }
    return 0;
}
```