

# Fondamenti di Informatica T-1

## Modulo 2

---

### Obiettivi di questa lezione

---

1. Valutazione in cortocircuito
2. If e if innestati
3. Switch
4. Cicli

# Valutazione in cortocircuito

---

In C, le **espressioni booleane sono valutate in cortocircuito** (short circuit 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

---

```
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 1

(valutazione in cortocircuito)

---

```
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);
}
```

Eseguire questo esercizio “sulla carta”...Che cosa verrà stampato a video? Quanto varranno le variabili le variabili al termine del programma?

# Esercizio 1

(if innestati)

---

## Esercizio 1 – 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
- Suggerimento: utilizzare if innestati

# Esercizio 1 - Soluzione

(if innestati)

---

```
int main() {
    double primo, secondo, terzo;
    printf("Lunghezze lati triangolo? ");
    scanf("%lf%lf%lf", &primo, &secondo, &terzo);
    if (primo == secondo)
    {
        if (secondo == terzo)
            printf("Equilatero\n");
        else
            printf("Isoscele\n");
    }
    else
    {
        if (secondo == terzo || primo == terzo)
            printf("Isoscele\n");
        else
            printf("Scaleno\n");
    }
    return 0;
}
```

## Esercizio 2

(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, utilizzare degli if innestati

# Esercizio 2 - Soluzione

## (if innestati)

---

```
#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 2 - Soluzione

## (if innestati)

---

```
...
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 1

## (switch)

---

### Stampa di voti

- Realizzare un programma che legge da input un voto (carattere tra 'A' ed 'E') e ne stampa il significato (A significa ottimo, E significa gravemente insuff., etc. etc.)

## Esercizio 1 - Soluzione

### (switch)

---

```
#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 2

(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"
- Suggerimento: utilizzare il costrutto switch
  - il comportamento del programma è determinato in base al valore del carattere "operatore"

## Esercizio 2 - Soluzione

(switch)

---

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

# Esercizio 3

(switch)

---

- Progettare e Codificare in C un programma che permetta di stabilire la stagione corrispondente al mese inserito come intero dall'utente.
  - Esempio: se l'utente digita 1 (Gennaio) il programma deve stampare Inverno.
- Qualora il mese sia Marzo, Giugno, Settembre e Dicembre, tutti mesi a cavallo di due stagioni, si richieda all'utente di specificare anche il giorno. Se il giorno e' compreso tra 1 e 20 si considera la stagione precedente altrimenti quella successiva.

## Esercizio 3 - Soluzione

(switch)

---

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int mese, giorno;
    printf("Inserisci il numero del mese: ");
    scanf("%d", &mese);
    switch (mese) {
        case 1:
        case 2: printf("Inverno!\n");
                break;
        case 4:
        case 5: printf("Primavera!\n");
                break;
        case 7:
        case 8: printf("Estate!\n");
                break;
        case 10:
        case 11: printf("Autunno!\n");
                 break;
    }
    ...
}
```



# Esercizio 3 - Soluzione

## (switch)

---

```
case 3: printf("Inserire anche il giorno: ");
        scanf("%d", &giorno);
        if (giorno >= 1 && giorno <= 20) printf("Inverno!\n");
        else printf("Primavera!\n");
        break;
case 6: printf("Inserire anche il giorno: ");
        scanf("%d", &giorno);
        if (giorno >= 1 && giorno <= 20) printf("Primavera!\n");
        else printf("Estate!\n");
        break;
case 9: printf("Inserire anche il giorno: ");
        scanf("%d", &giorno);
        if (giorno >= 1 && giorno <= 20) printf("Estate!\n");
        else printf("Autunno!\n");
        break;
case 12: printf("Inserire anche il giorno: ");
         scanf("%d", &giorno);
         if (giorno >= 1 && giorno <= 20) printf("Autunno!\n");
         else printf("Inverno!\n");
         break;
default: printf ("Valore errato!");

system("PAUSE");
return 0; }
```

•17

## Esempio

### (cicli)

---

### Echo di caratteri

- Realizzare un programma che legga ciclicamente da input un carattere e lo stampi a video
- Il programma deve terminare quando l'utente inserisce il carattere '0'

# Esempio

## (cicli)

---

```
#include <stdio.h>

int main(void)
{
    char c;
    do
    {
        scanf("%c", &c);
        printf("%c", c);
    } while(c > 0);
    return 0;
}
```

# Esercizio 1

## (cicli)

---

- 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 1 - Soluzione

## (cicli)

---

```
#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 2

## (cicli)

---

### Calcolo di un logaritmo in base qualunque

- Realizzare un programma che chieda all'utente due numeri razionali, uno detto base B e l'altro detto valore X (a cui applicare il logaritmo)
- Se e solo se **B e X sono entrambi positivi**, il programma deve fornire come risposta il logaritmo in base B di X...
- ...altrimenti deve stampare un messaggio di errore

# Esercizio 2

(cicli)

- Si rammenti che, data una funzione logaritmica in base k, un logaritmo in una base diversa B è così definito:

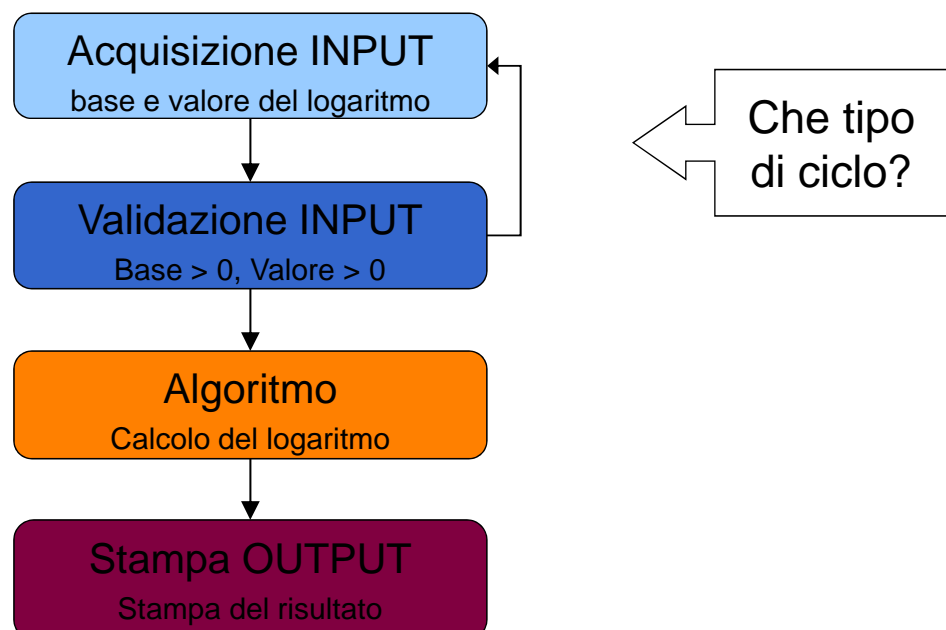
$$\log_B(x) = \frac{\log_k(x)}{\log_k(B)}$$

- Approccio bottom-up: posso sfruttare del codice già fatto?
- Si consideri l'uso delle funzioni  $\log(X)$  (logaritmo naturale) e  $\log_{10}(X)$ , disponibili in **math.h**

# Esercizio 2

(cicli)

Schema di soluzione



# Esercizio 2 - Soluzione

## (cicli)

---

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    double b, x, result;
    do //ACQUISIZIONE CONTROLLATA b
    {
        printf("Base (>0 e !=1): ");
        scanf("%lf",&b);
    }
    while(b <= 0 || b == 1);
    do //ACQUISIZIONE CONTROLLATA x
    {
        printf("Valore (>0): ");
        scanf("%lf",&x);
    }
    while(x <= 0);
```

# Esercizio 2 - Soluzione

## (cicli)

---

```
//Calcolo del risultato
result = log(x)/log(b);

//Stampa del risultato
printf("log %lf ( %lf ) = %lf\n", b, x, result);

return 0;
}
```

## Esercizio 3

### (cicli)

---

- Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi terminati dal valore 0 (uno su ogni linea) e stampi il prodotto degli interi positivi e la somma dei negativi.
- Codificare il programma in C
- Procedere alle operazioni di Compile, Link e correggere eventuali errori.
- Seguire l'esecuzione del programma con l'uso del debugger.

## Esercizio 3 - Soluzione

### (cicli)

---

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

int main ()
{
    int somma, prod, num;
    somma = 0;
    prod = 1;

    do {
        scanf("%d", &num);
        if (num > 0)
            prod = prod * num;
        else if (num < 0)
            somma = somma + num;
    } while (num != 0);

    printf("Somma: %d\n", somma);
    printf("Prodotto: %d\n", prod);

    system("PAUSE");
    return 0;
}
```

## Esercizio 4

### (cicli)

---

- Progettare un algoritmo che legga da terminale una sequenza di interi positivi e negativi terminati dal valore 0 (uno su ogni linea) e stampi la media degli interi positivi.
- Codificare il programma in C e scriverlo in **voto.c**
- Procedere alle operazioni di Compile, Link e correggere eventuali errori.
- Seguire l'esecuzione del programma con l'uso del debugger.

## Esercizio 4 - Soluzione

### (cicli)

---

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int somma = 0, num, cont = 0;
    float media;
    printf("Inserisci un numero: ");
    scanf("%d", &num);
    while (num != 0) {
        if (num > 0) {
            somma = somma + num;
            cont++;
        }
        printf("Inserisci un numero: ");
        scanf("%d", &num);
    }
    if (cont==0)
        printf("Non hai inseriti alcun numero!\n");
    else {
        media = ((float) somma) / cont;
        printf("Media: %f\n", media);
    }
    system("PAUSE");
    return 0; }
```

## Esercizio 5

### (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 5 – Soluzione 1

### (cicli)

---

```
#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 5 – Soluzione 2 (cicli)

---

```
#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 6 (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 6 - Soluzione

### (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 7

### (cicli)

---

Progettare e Codificare in C un programma che permetta di

- Chiedere all'utente quanti numeri vuole inserire
- Leggere i numeri inseriti dall'utente e calcolare la somma dei fattoriali
- Esempio: L'utente vuole inserire 3 numeri:
- 4, 3, 6
- Il programma deve calcolare  $4! + 3! + 6! = 750$

## Esercizio 7 - Soluzione

### (cicli)

---

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

int main(void) {
    int limit, num;
    int i, j;
    int part, tot;
    printf("Quanti numeri vuoi inserire? ");
    scanf("%d", &limit);

    tot = 0;
    for (i=0; i<limit; i++) {
        printf("Inserisci il numero %d:", i+1);
        scanf("%d", &num);
        part = 1;
        for (j=num; j>0; j--)
            part=part*j;
        tot = tot + part;
    }
    printf("Totale: %d\n", tot);
    system("PAUSE");
    return (0); }
```

## Esercizio 8

### (cicli)

---

Progettare e Codificare in C un programma che permetta di controllare i dati di input immessi dall'utente.

- In particolare se l'utente inserisce un intero N compreso tra 1 e 10, il programma deve stampare a video il valore  $N^N$
- Se l'intero N e' compreso tra 11 e 20, il programma deve stampare a video la somma  $1 + 2 + 3 + \dots + N$

$$\sum_{j=1}^N j$$

- Altrimenti deve dare un segnale di errore.

## Esercizio 8 - Soluzione

### (cicli)

---

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

int main () {
    int num, i, result;
    printf("Inserisci un numero: ");
    scanf("%d", &num);

    if (num >= 1 && num <= 10) {
        result = 1;
        for (i=1; i<=num; i++)
            result = result * num;
    }
    else
        if (num >= 11 && num <= 20) {
            result = 0;
            for (i=1; i<=num; i++)
                result = result + i;
        }
    else printf("Errore!\n");
    printf("Valore calcolato: %d\n", result);
    system("PAUSE");
    return 0; }
```

## Esercizio 9

### (cicli)

---

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 9 - Soluzione

### (cicli)

```
#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 10

### (cicli)

### CALCOLO DEGLI INTERESSI BANCARI

Si progetti in C un programma che legge un float, rappresentante un ammontare di euro; di seguito il programma deve leggere un tasso d'interesse (in percentuale), ed un numero di anni.

Il programma deve stampare, in uscita, per ogni anno, come l'ammontare cresce con gli interessi. Si ricordi che l'interesse si calcola con la seguente formula:

$$C_{fin} = C_{in} * \left(1 + \frac{r}{100}\right)^N$$

Dove  $C_{fin}$  è il capitale finale,  $C_{in}$  è quello iniziale,  $r$  è l'interesse, e  $N$  rappresenta il numero di anni in cui si applicano gli interessi.

*Continua >*

# Esercizio 10

## (cicli)

---

Supponiamo che il capitale iniziale sia di 1000.0 €, con un tasso del 3%, per un periodo di 3 anni. L'output stampato deve avere all'incirca questo aspetto:

Capitale iniziale: 1000.00€

Dopo 1 anno: 1030.00 €

Dopo 2 anni: 1060.90 €

Dopo 3 anni: 1092.73 €

Suggerimento: scomporre il problema in sotto-problemi più semplici...

- All'inizio è necessario chiedere all'utente alcuni parametri (quali?)
- Per ogni anno è necessario calcolare il capitale finale
- Il capitale finale viene calcolato tramite la formula

---

## Esercizio 10 - Soluzione

### (cicli)

---

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

int main(void)
{
    float c_in;
    float c_finale;
    float tasso, tasso_perc, tasso_tot, tasso_anni;
    int anni;
    int i, j;

    printf("Inserisci capitale iniziale: ");
    scanf("%f", &c_in);

    printf("Inserisci tasso d'interesse: ");
    scanf("%f", &tasso);

    printf("Inserisci numero di anni: ");
    scanf("%d", &anni);
```

...

*Continua >*

## Esercizio 10 - Soluzione (cicli)

---

```
...

printf("Capitale iniziale: %6.2f\n", c_in);
for (i=1; i<=anni; i++)
{
    tasso_anni = 1;
    tasso_perc = tasso/100;
    tasso_tot = 1 + tasso_perc;
    for (j=1; j<=i; j++)
        tasso_anni=tasso_anni*tasso_tot;
    c_finale = c_in * tasso_anni;
    printf("Capitale dopo %d anni: %6.2f\n", i, c_finale);
}

system("PAUSE");

return (0);
}
```

## Esempio su cicli iterativi

---

### Sequenze di '0' e '1'

- Realizzare un programma che prende in input una sequenza di caratteri '0' e '1' e **conta la lunghezza della più lunga sotto-sequenza di '0' di fila**
- L'inserimento della sequenza **termina** quando si inserisce un **carattere diverso da '0' e '1'**
- A quel punto, si stampa a video il valore trovato

# Esempio su cicli iterativi

---

```
char bit;
int cont = 0, maxlung = 0;
printf("Inserisci la sequenza\n");
do
{
    bit = getchar();
    getchar(); //consumo il newline
    if (bit == '0')
    {
        cont++;
        if(cont > maxlung)
            maxlung = cont;
    }
    else
    {
        cont = 0;
    }
}
while(bit == '0' || bit == '1');
printf("Lunghezza massima sotto-sequenza di 0: %d\n", maxlung);
```

## Esercizio 11

(cicli)

---

### Sequenze di cifre

- Realizzare un programma che prenda in input una **sequenza di cifre (tra 1 e 9)** e calcoli la **somma massima fra le sotto-sequenze di cifre non decrescenti**
- Il programma termina quando viene inserito lo 0

Esempio:

2	2	4	5	3	9	3	1	5	0
13				12		3	6		



# Esercizio 11 - considerazioni

## (cicli)

---

- Ci mettiamo nell'ipotesi di non dover controllare la correttezza del carattere letto
- Come effettuo il passaggio **carattere-cifra numerica**?
  - Quanto fa '8' - '0' (considerato come intero)?
- Di che valori devo tenere traccia?
  - Devo accorgermi di quando il valore appena letto è più piccolo del precedente
    - In tal caso, devo confrontare la somma corrente con quella massima, e ripartire con una nuova somma

# Esercizio 11 - Soluzione

## (cicli)

---

```
char cifra;
int curVal = 0, oldVal = 0, somma = 0, sommaMax = 0;
do
{
    scanf("%c", &cifra);
    getchar();
    curVal = cifra - '0';
    if (curVal < oldVal)
    {
        if (somma > sommaMax)
            sommaMax = somma;
        somma = curVal;
    }
    else
        somma = somma + curVal;
    oldVal = curVal;
}
while (curVal != 0);
printf("Massima somma trovata: %d\n", sommaMax);
```