

Fondamenti di Informatica e Laboratorio T-AB  
Ingegneria Elettronica e Telecomunicazioni

---

# Lab 04

# Programmazione Strutturata

# 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 1

---

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 2

---

## Stampa di voti

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

# Esercizio 2 – 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 3

---

## **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 3 - 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 4

---

## **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 4 - 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 4 - 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 5

---

## 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 5 - 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 6

---

## 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 6

---

```
#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;
}
```

# Esercizio 7

---

- 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 è compreso tra 1 e 20 si considera la stagione precedente altrimenti quella successiva.

# Esercizio 7 - Soluzione

---

```
#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 7 - Soluzione

---

```
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!");
return 0; }
```

# Esercizio 8

---

- 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 8 - 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 9

---

Si realizzi un programma che, partendo da una base **a** ed un esponente **i**, calcoli la funzione potenza:

$$a^i$$

# Esercizio 9 - Soluzione

---

```
#include <stdio.h>

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

    int prod;

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

    for (j=0; j < i; ++j) {
        prod = prod*a;
    }

    printf("Valore della potenza: %d", prod);
    return 0;
}
```

# Esercizio 10 – 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 10 - 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 10 - 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 11

---

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 11

---

```
#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 12 - 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 12 - 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 13

---

- 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 14

---

- 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.
- Seguire l'esecuzione del programma con l'uso del debugger.

# Esercizio 14 - Soluzione

---

```
#include <stdio.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);

    return 0;
}
```

# Esercizio 15

---

- 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.
- Seguire l'esecuzione del programma con l'uso del debugger.

# Esercizio 15 - Soluzione

---

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

int main () {
    int somma, num, cont;
    float media;

    somma = 0;
    cont = 0;
    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) media = ((float) somma) / cont;
    printf("Media: %f\n", media);
    return 0; }
```

# Esercizio 16

---

- 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 16 - Soluzione

---

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

    return 0; }
```

# Esercizio 17

---

## 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

# Esercizio 17

---

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

# Esercizio 18

---

## Sequenze di cifre

- Realizzare un programma che prende in input una sequenza di cifre (tra 1 e 9) e calcola la somma massima fra le sotto-sequenze di cifre non decrescente
- Il programma termina quando viene inserito lo 0
- Esempio:

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

# Esercizio 18

---

- 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 18

---

Le cifre sono trattate come caratteri in questa soluzione

```
#include <stdio.h>
int main() {
    char cifra;
    int cur_val = 0, old_val = 0, somma = 0, somma_max = 0;
    do {
        scanf("%c", &cifra);
        getchar();
        cur_val = cifra - '0';
        if(cur_val < old_val)
        {
            if( somma > somma_max) somma_max = somma;
            somma = cur_val;
        }
        else somma = somma + cur_val;
        old_val = cur_val;
    } while(cur_val != 0);
    printf("Massima somma trovata: %d\n", somma_max);
    return 0;
}
```

# Esercizio 19

---

Si scriva un programma che legga da utente una sequenza di interi terminata dal valore 0, moltiplichi i valori positivi pari e sommi i positivi dispari scartando i numeri negativi.

Si scriva il programma usando il ciclo **do..while**

In un secondo tempo si modifichi il programma usando il ciclo **while**

Si sviluppi il programma usando un solo ciclo

*Continua >*

# Esercizio 19- do..while un ciclo

---

```
#include <stdio.h>
int main()
{
int num, S_dispari=0, P_pari = 1, cont_pari = 0;

do{ printf("Inserisci num positivo - 0 per terminare \n");
scanf("%d", &num);
if (num>0)
    {if (num %2 == 0)
        {cont_pari++;
        P_pari = P_pari * num;}
    else S_dispari = S_dispari + num;}
} while(num != 0);
if (cont_pari >0)
printf("Prodotto numeri pari = %d \n", P_pari);
printf("Somma numeri dispari = %d", S_dispari);
return 0;
}
```

# Esercizio 19- do..while due cicli

---

```
#include <stdio.h>
int main()
{int num, S_dispari=0, P_pari = 1, cont_pari = 0;
do{
    do{
        printf("Inserisci num positivo - 0 per terminare \n");
        scanf("%d", &num);}
    while(num < 0);
    if (num>0)
        {if (num %2 == 0)
            {cont_pari++;
             P_pari = P_pari * num;}
         else S_dispari = S_dispari + num;}
} while(num != 0);

if (cont_pari >0)
printf("Prodotto numeri pari = %d \n", P_pari);
printf("Somma numeri dispari = %d", S_dispari);
return 0;}
```

# Esercizio 20

---

## 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 20

---

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 €**

# Esercizio 20 - Soluzione

---

```
#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 20 - Soluzione

---

...

```
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);
}

return (0);
}
```

# Esercizio 21

---

Si scriva un programma che legga da utente una sequenza di al massimo 10 valori positivi (si scartino i negativi). L'utente può terminare prima inserendo lo 0.

Il programma deve stampare

- 1.il numero di valori positivi letti
- 2.il numero di valori scartati
- 3.la somma di tutti i valori positivi
- 4.L'elemento maggiore della sequenza dei positivi
- 5.L'elemento minore della sequenza dei positivi

*Continua >*

# Esercizio 21- Soluzione

---

```
#include <stdio.h>
int main()
{
int num, S_pos=0, N_pos = 0, N_scartati = 0, max, min ;
    printf("Inserisci un numero positivo - 0 per terminare \n");
    scanf("%d", &num);

while (num != 0 && N_pos<10)
    { if (num > 0)
        {if (N_pos == 0)
            {max = num;
            min = num;
            }
        N_pos++;
        S_pos = S_pos + num;
        if (num > max)
            max = num;
        else if (num < min)
            min = num;}
    else N_scartati ++;
    printf("Inserisci un numero positivo - 0 per terminare \n");
    scanf("%d", &num);
}
```

# Esercizio 21- Soluzione

---

```
printf("Numero positivi %d\n", N_pos);  
printf("Numero scartati %d\n", N_scartati);  
printf("Max positivi = %d \n", max);  
printf("Min positivi = %d \n", min);  
printf("Somma positivi = %d \n", S_pos);  
  
return 0;  
}
```