

Definizione di utente e di programmatore

L'utente è qualsiasi persona che usa il computer anche se non è in grado di programmarlo. **Il programmatore è l'uomo che programma un computer.**

Definizione di istruzione elementare

E' una istruzione non ulteriormente scomponibile e di semplice comprensione per la CPU se non è ambigua.

Esempi di istruzioni elementari:

- 1) istruzioni aritmetico - logiche.** Sono quelle eseguite dall'ALU nelle operazioni aritmetiche (+, -, / e *) e nelle operazioni logiche (and, or, not)
- 2) istruzioni di trasferimento.** Sono quelle usate nel trasferimento dei dati all'interno della memoria centrale (RAM) e tra questa ed i registri della CPU
- 3) istruzioni di input / output.** Sono quelle che consentono l'immissione e l'emissione dei dati nella e dalla memoria centrale (RAM) attraverso l'uso delle periferiche di input e di output
- 4) istruzioni di controllo.** Sono quelle che guidano lo svolgimento dell'elaborazione controllando l'ordine di esecuzione delle istruzioni elementari e sono eseguite dall'unità di controllo (CU).

Definizione di linguaggio di programmazione

Esso è il linguaggio usato dal programmatore per comunicare con il computer. Il primo linguaggio usato è stato quello binario, in seguito fu l'assembler (un misto di codice binario a parole utente come: add (per fare la somma), molt (per fare le moltiplicazioni)), ecc. Negli anni sono nati molti linguaggi di programmazione sempre più vicini al modo di pensare del programmatore.

Esempi di linguaggi di programmazione inventati dal programmatore negli anni

- **ForTran** (Formula Translation) nato nel 1954 per scopi scientifici e per calcoli matematici
- **COBOL** (Common Business Oriented Language) nato nel 1960 per scopi commerciali
- **PASCAL** nato nel 1971 ad opera di N. Wirth progettato come linguaggio per l'insegnamento nelle scuole della programmazione strutturata (ancora in uso in qualche ITIS)
- **BASIC** (Beginner's All Purpose Symbolic Instruction Code) **nato nel 1980** per avere un linguaggio che fosse molto semplice da imparare anche per i principianti
- **Visual Basic** nato alle fine degli anni 90 e poi il **Visual Basic . Net** nato nel 2003. Entrambi usano il lessico (parole chiavi) base del linguaggio Basic, ma con l'aggiunta di molte altre funzioni e procedure di libreria.
- **C** nato nel 1972 è usato per tutti gli scopi. E' di tipo strutturato e molto usato in didattica anche se si tratta di un linguaggio non particolarmente intuitivo per i principianti. Evoluzioni del C sono: **C++** (nato nel 1983), **Java** (nel 1995) e **C#** (nato nel 2000) e aggiornato alla versione 7 conenuto nel pacchetto visual studio del 2017

Definizione di programma

Un programma è un sequenza finita di istruzioni elementari che indichino al computer le operazioni da compiere e i dati di ingresso su cui operare, scritte in un linguaggio di programmazione.

Definizione di algoritmo.

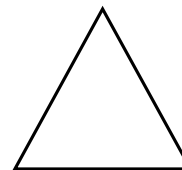
Un algoritmo è una sequenza ordinata di azioni o istruzioni che permettono di risolvere un problema nel modo più efficiente possibile (cioè col minor numero di istruzioni). Le istruzioni dell'algoritmo, inoltre, devono soddisfare ai seguenti requisiti:

- 1. elementari**, cioè non ulteriormente scomponibili
- 2. in numero finito**, nel senso che l'algoritmo deve avere un punto di inizio e un punto di fine

3. **non ambigue**, cioè che non lascino dubbi di interpretazione alla CPU che dovrà eseguirle.
4. **deterministiche**, cioè devono garantire che i risultati ottenuti siano sempre gli stessi a prescindere dalla CPU che le esegue. Ad esempio: l'istruzione "moltiplica 3 per 4" produce sempre il medesimo risultato, anche se è eseguita su CPU diverse.

Problema1. Trovare l'algoritmo che calcola l'area del triangolo

1. inizio
2. dammi valore della base
3. dammi valore della altezza
4. calcola il prodotto: base per l'altezza
5. dividi il prodotto per due ed otterrai il valore dell'area del triangolo
6. stampa a video l'area calcolata
7. fine



Variabili e Costanti

In tutti i linguaggi di programmazione il programmatore ha necessità di memorizzare temporaneamente alcuni valori del programma. Tali valori sono memorizzati nelle celle della RAM.

La variabile è una cella della RAM avente le seguenti caratteristiche:



Inoltre, la variabile è individuata da:

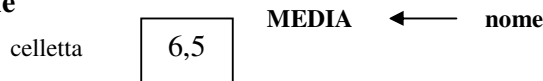
- **un nome scelto a piacere dal programmatore composto da lettere e numeri** ma che non deve mai iniziare con un numero come ad es: **numero, a1, a2**, somma, media, nome, ecc.
- **un valore modificabile nel tempo**, a seconda che si verifichino o meno condizioni, eventi, ecc. da qui il nome variabile, **che può essere: un numero intero positivo o negativo, un numero con la virgola, un carattere, una parola, un valore booleano vero e falso, ecc.**
- **un tipo**, cioè l'insieme dei valori che essa può assumere, ad es. **intero, reale, stringa, ecc**
- **di ingresso, di uscita oppure ne di ingresso e ne di uscita (detta anche di lavoro)**
- **descrizione:** spiegazione breve del suo ruolo nell'algoritmo.

Esempio1 - variabile intera



In tal caso **A** è il nome scelto per la variabile, **6** è il suo valore in un certo istante e il suo tipo è quello intero. **In C++ indicata con int.**

Esempio2 - variabile decimale



In tal caso **MEDIA** è il nome scelto per la variabile, **6,5** è il suo valore e il suo tipo è quello decimale. **In C++ indicata con float o double.**

Esempio3 - variabile stringa



In tal caso **COLORE** è il nome scelto per la variabile, **nero** è il suo valore e il suo tipo è quello stringa. **In C++ indicata con char se costituita da un solo carattere e con string se sequenza.**

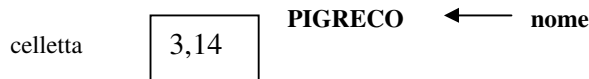
Esempio4 – variabile booleana

In tal caso MAGGIORENNE è il nome scelto per la variabile, vero è il suo valore e il suo tipo è quello booleana



In C++ una variabile booleana si chiamerà di tipo Bool e potrà assumere valori: true e false

La costante è una celletta della RAM in cui è possibile memorizzare temporaneamente i valori che non variano nel tempo, cioè che sono costanti.



In tal caso PIGRECO è il nome scelto per la costante, invece 3,14 è il suo valore. Si dice che la costante PIGRECO è di tipo “decimale”, poiché contiene un numero con la virgola

L’istruzione di assegnazione delle variabili

Essa permette di attribuire ad una variabile un valore

Esempi di assegnazioni con costanti e variabili:

- raggio = 3,14
- voto = 7
- targa = “dx112ed”
- sposato = true
- conta = conta +1

La tabella delle variabili

Essa raggruppa in sintesi le variabili usate dal programma specificando per ognuna di esse:

- una descrizione sintetica dei valori che rappresenta
- il tipo di valori che può assumere
- se è una variabile di input o di output
- se è una variabile di lavoro, cioè né di input e né di output.

Esempio1. Tabella delle variabili dell’algoritmo area del triangolo (visto in precedenza)

Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
b	Base del triangolo	Numerico intera	X		
h	Altezza del triangolo	Numerico intera	X		
a	Area del triangolo	Numerica decimale		X	

Problema2. Trovare l’algoritmo che calcoli l’area del rettangolo. Prima dell’algoritmo disegnare la tabella delle variabili.

Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
b	Base del rettangolo	Numerico intera	X		
h	Altezza del rettangolo	Numerico intera	X		
a	Area del rettangolo	Numerica intera		X	

Algoritmo

1. Inizio
2. Dammi il valore della base b
3. Dammi il valore della altezza h
4. Calcola il prodotto $A = b \times h$
5. Stampa a video la sua area A
6. Fine



Il diagramma a blocchi o flowchart o diagramma di flusso

Esso è il metodo usato dai programmatori per trovare l'algoritmo (cioè la soluzione) a problemi complessi. Esso consiste nel descrivere la sequenza di istruzioni elementari di un algoritmo in modo grafico attraverso l'uso di simboli detti blocchi, collegati tra loro da frecce orientate che indicano il flusso delle azioni.

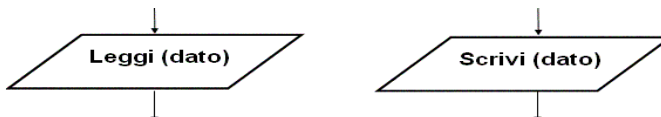
I blocchi principali sono quattro: ovale, parallelogramma, rettangolo e rombo.

Ciascun blocco ha un proprio significato. I blocchi sono: blocco di inizio o di fine, blocco di lettura o di scrittura, blocco di elaborazione, blocco di decisione o di condizione. Ma vediamoli più nel dettaglio:

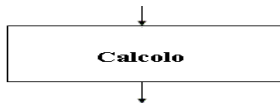
1. Il blocco a forma di ovale. Esso si usa per indicare il punto di partenza e quello di terminazione dell'algoritmo. Dal blocco "inizio" parte una sola freccia che raggiunge la prima azione dell'algoritmo e verso "inizio" non arriva alcuna freccia. Invece, verso il blocco "fine" arrivano una o più frecce (in questa fig. solo una), ma da esso non ne parte nessuna.



2. Il blocco a forma di parallelogramma. Esso si usa per indicare un'operazione di immissione di dati (detta operazione di lettura o di input) o un'operazione di emissione di dati (detta anche operazione di scrittura o di output). Verso il blocco, in genere, ci sono una o più frecce di entrata (in questa fig. solo una) e dal blocco c'è una sola freccia di uscita.



3. Il blocco a forma di rettangolo. Esso si usa per indicare un'azione di calcolo. Verso il blocco, in genere, ci sono una o più frecce di entrata (in questa fig. solo una) e dal blocco c'è una sola freccia di uscita.



4. Il blocco a forma di rombo. Esso si usa per indicare un'operazione di confronto tra due dati, per stabilire se la condizione in esso inserita è vera o è falsa. Verso il blocco c'è una sola freccia di entrata e dal blocco ci sono solo due frecce di uscita in corrispondenza delle quali si trovano indicazioni del tipo si/no, vero/falso o V/F.

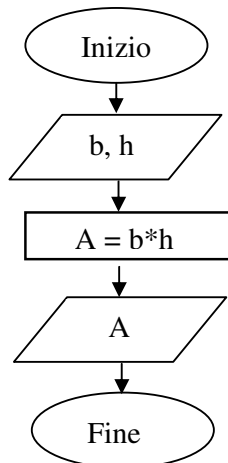


Le frecce di direzione

Esse sono frecce orientate e sono usate per collegare i vari blocchi tra loro.

Esempio di diagramma a blocchi

A partire dall'algoritmo del calcolo dell'area del rettangolo, dedurre il diagramma a blocchi seguente:



Altri simboli usati ma non sempre sono:



un piccolo quadrato: per il punto di connessione tra frecce orientate	un rettangolo con 2 barre laterali per i sottoprogrammi: function e procedure	un esagono per la struttura ciclica con contatore, ad es. per i che va da 1 a 100
---	---	---

Problema3: Dati due numeri A e B, trovare l'algoritmo che calcoli la loro somma. Disegnare la tabella delle variabili e il diagramma a blocchi.

1) la tabella delle variabili di input, di output e di lavoro è la seguente:

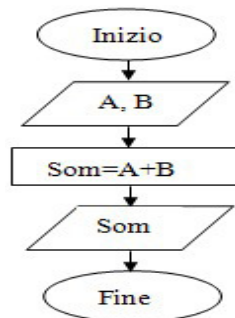
Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
A	Primo numero	Numero intero	X		
B	Secondo numero	Numero intero	X		
Som	Somma dei 2 numeri	Numero intero		X	

2) Individuazione dell'algoritmo per passi

Algoritmo per passi

1. inizio
2. dammi il valore del primo numero A
3. dammi il valore del secondo numero B
4. somma i due numeri A e B
5. visualizza la somma Som
5. fine

3) Individuazione del diagramma a blocchi



Cosa serve per realizzare un programma in C/C++?

Bisogna eseguire nell'ordine i seguenti passi:

- 1) Dato un problema, disegnare la tabella delle variabili (spiegato)
- 2) Individuare l'algoritmo risolutivo per passi (detto anche pseudo codifica) (spiegato)
- 3) Individuare il diagramma a blocchi (detto anche flowchart) (spiegato)
- 4) Farne il test che dimostri la bontà dell'algoritmo e delle sue possibili correzioni
- 5) Tradurre l'algoritmo in codice C++, cioè dedurre dall'algoritmo il programma.

Un programma in C/C++

Esso è una sequenza finita di istruzioni scritte in linguaggio C++, ricavate dal programmatore C++, a partire dall'algoritmo su carta. Tale programma è detto anche programma sorgente.

Un compilatore C/C++

Esso è un programma che permette di tradurre un programma sorgente .c oppure .cpp in programma oggetto (file .obj) scritto in linguaggio binario.

Un linker C/C++


Esso è un programma che permette di collegare il programma oggetto creato dal compilatore con le librerie e sottoprogrammi del linguaggio, richieste nel programma e necessarie alla sua esecuzione ovvero necessarie per creare il programma eseguibile. (file.exe)

Requisiti per la creazione di un programma in C/C++

Per la creazione di un programma in C/C++ basta avere un editor come ad es: il blocconote di windows oppure l'editor colorato, opensource e portatile notepad++, scaricabile dal sito <https://notepad-plus-plus.org/download/v7.5.4.html> Poi bisogna salvare il file di testo creato con estensione .c oppure .cpp

Esempio di compilatore gratuito C/C++

Un compilatore C/C++ è ad es. il DEV-C++ scaricabile dal sito <http://www.bloodshed.net/dev>
Esso permette, da solo, di editare, compilare, linkare ed eseguire un programma .c o .cpp

	<pre>#include <iostream> using namespace std; int main() { int a, b, s; /* a,b: sono i numeri, s è la somma */ cout<<"A="; cin>>a; cout<<"B="; cin>>b; s=a+b; cout<<"\nLA SOMMA E': Som= "<<s; return 0; }</pre>
---	--

Alcune istruzioni presenti sempre nei programmi scritti in linguaggio C++

1) **#include <iostream>**. Ogni programma in C++ deve iniziare con l'istruzione #include <iostream>. Essa e tutte le istruzioni che iniziano per il simbolo di # sono dette direttive per il compilatore. Essa in particolare, serve per poter usare nel programma le funzioni per l'input (cin>>), per l'output (cout<<), la funzione main(), ecc.

2) **using namespace std;** letteralmente significa poter usare lo spazio dei nomi (nombrespace) necessario per evitare che due funzioni (o sottoprogrammi) abbiano lo stesso nome e quindi ricevere

un errore in compilazione. Nei nostri programmi utilizzeremo "using namespace std" solo per evitare di dover scrivere `std::cin` o `std::cout` funzioni al posto di `cin` e `cout` largamente usate nei nostri programmi

3) **int main ()** indica il punto in cui inizia l'esecuzione di un qualsiasi programma C++. E' indispensabile che ogni nostro programma in c++ sia all'interno della funzione **main**.

4) **return 0;** tale istruzione fa terminare la funzione main() e ritorna quello che è indicato di seguito, nel nostro caso **0**. Questo è il modo normale di terminare un programma la cui esecuzione è avvenuta senza errori. Se dovesse ritornare ad es. il valore -1, non si potrebbe eseguire il programma.

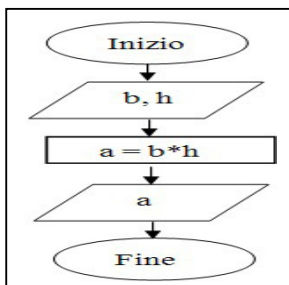
5) **// o /* */** Questa è una riga di commento e non ha alcun effetto sul comportamento del programma. Essa può essere usata dal programmatore per includere nel codice del programma alcune brevi spiegazioni ed osservazioni.

Esempio1 – Calcolo dell’algoritmo del calcolo dell’area del rettangolo.

Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
b	Base del rettangolo	Numerico intera	X		
h	Altezza del rettangolo	Numerico intera	X		
a	Area del rettangolo	Numerica intera		X	

Algoritmo

1. Inizio
2. Dammi il valore della base b
3. Dammi il valore della altezza h
4. Calcola il prodotto $A = b \times h$
5. Stampa a video la sua area A
6. Fine



```

#include <iostream>
using namespace std;
int main() {
    int a, b, h;
    /* a è l'area, b= base; h = altezza */
    cout<<"CALCOLO AREA \n";
    cout<<"B="; cin>>b;
    cout<<"H="; cin>>h;
    a=b*h;
    cout<<"\nL'AREA E' : A= "<<a;
    return 0; }
    
```

Problema4: Dati due numeri A e B, calcolarne la somma.

Per la realizzazione di questo programma seguiamo i suddetti 5 passi:

1) la tabella delle variabili di input, di output e di lavoro è la seguente:

Tabelle delle variabili di input, output e di lavoro					
Nome	Descrizione	Tipo	Input	Output	Lavoro
A	Primo numero	Numero intero	X		
B	Secondo numero	Numero intero	X		
Som	Somma dei 2 numeri	Numero intero		X	

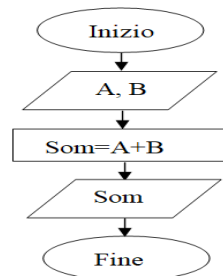
2) Individuazione dell'algoritmo per passi

Algoritmo per passi

1. inizio
2. dammi il valore del primo numero A
3. dammi il valore del secondo numero B
4. somma i due numeri A e B
5. assegna la somma del punto 4 a som
6. visualizza a video la somma Som
5. fine

3) individuazione del diagramma a blocchi

Algoritmo col diagramma a blocchi



Problema5: Dati un numero A, calcolare il suo quadrato QA e doppio DA e li stampi a video

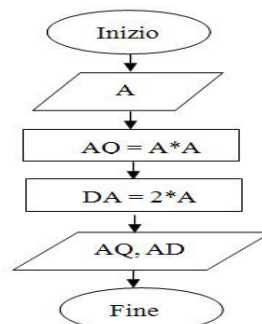
1) Tabelle delle variabili di input e output

Nome	Descrizione	Tipo	Input	Output	Lavoro
A	Primo numero	Numero intero	X		
QA	Quadrato di A	Numero intero		X	
DA	Doppio di A	Numero intero		X	

2) Algoritmo per passi

1. inizio
2. dammi il valore del primo numero A
3. calcola il suo quadrato, $AQ=A*A$
4. calcola il suo doppio, $DA=2*A$
5. visualizza AQ e DA
6. fine

3) Algoritmo col diagramma a blocchi



Esercizio: Date le basi maggiori e minori e l'altezza di un trapezio, calcolare l'area.

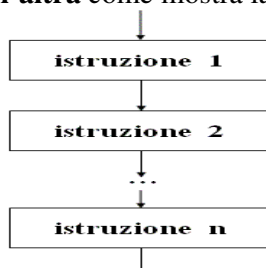
Per la realizzazione del programma usare la seguente tabella delle variabili:

1) Tabelle delle variabili di input, di output e di lavoro					
Nome	Descrizione	Tipo	Input	Output	Lavoro
B1	Base maggiore	Numero intero	X		
b2	base minore	Numero intero	X		
H	Altezza	Numero intero	X		
SOM	Somma (B1+b2)	Numero intero			X
P	Prodotto SOM*H	Numero intero			X
Area	Area del trapezio P/2	Numero decimale		X	

Per esercizio, svolgere da soli algoritmo, test sull'algoritmo e codice C++ e test sul codice.

Definizione di struttura di sequenza o sequenziale

In un algoritmo si parla di struttura di sequenza quando le istruzioni elementari dell'algoritmo sono una di seguito all'altra come mostra la figura:



Problema1: Scrivere un algoritmo che dati tre voti interi A, B e C letti in input, calcoli la media aritmetica e la stampi in output

1) Tabelle delle variabili di input, di output e di lavoro					
Nome	Descrizione	Tipo	Input	Output	Lavoro
A	1° voto	Numero intero	X		
B	2° voto	Numero intero	X		
C	3° voto	Numero intero	X		
SOM	Somma dei tre voti	Numero intero			X
MED	Media dei tre voti	Numero decimale		X	

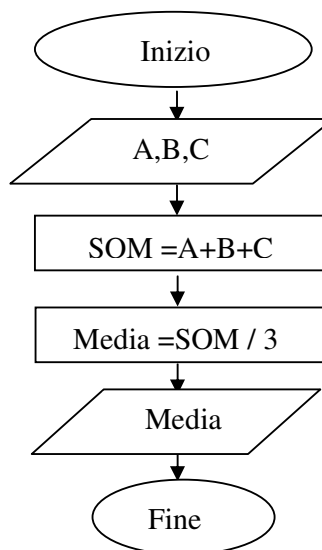
4) programma in C++

```

#include <iostream>
using namespace std;
int main() {
    int a, b; float med;
    /* med è la media, a, b sono i 2 numeri */
    cout<<"CALCOLO MEDIA \n";
    cout<<"A="; cin>>a;
    cout<<"B="; cin>>b;
    med=(a*b)/2;

    cout<<"\nLA MEDIA E' := "<<med;
    return 0;
}
  
```

3) Algoritmo col diagramma a blocchi



Problema2: Scrivere un algoritmo che dati due numeri X e Y, effettui lo scambio dei valori e stampi a video i nuovi valori assunti da esse, utilizzando una variabile di appoggio.



Capiamo il problema: sia x=10 e y=20

- 1) modo: sposto prima il contenuto di y in x, allora faccio x=y, quindi x sarà = 20, ma anche y = 20
- 2) modo: sposto prima il contenuto di x in y, allora faccio y=x, quindi y sarà = 10, ma anche x = 10

Bene in entrambi i modi ho ottenuto due valori uguali di x e di y e non è quello che volevo. Cosa devo fare? Prima di effettuare lo scambio, occorre conservare il valore di x o di y all'interno di un'altra variabile, detta temporanea, TEMP.

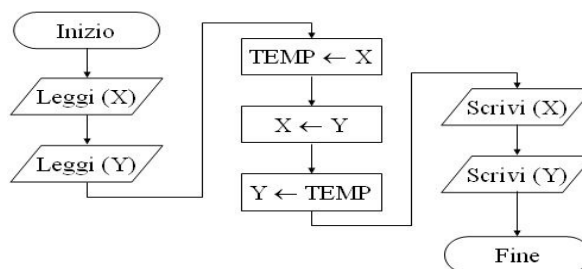
1) Tabelle delle variabili di input, di output e di lavoro

Nome	Descrizione	Tipo	Input	Output	Lavoro
X	Primo valore	Numero intero	X	X	
Y	Secondo valore	Numero intero	X	X	
TEMP	Variabile temporanea di appoggio	Numero intero			X

2) Algoritmo per passi

1. Inizio
2. Dammi i valori di X e Y
3. Assegna a TEMP il valore di X
4. Assegna ad X il valore di Y
5. Assegna ad Y il valore di TEMP
6. Stampa X e Y
8. Fine

3) Algoritmo col diagramma a blocchi



Problema3: Scrivi un algoritmo che dati in input quattro valori interi A, B, C e D, ruoti il loro valore di uno step in avanti in senso orario e scriva in output il loro valore modificato.



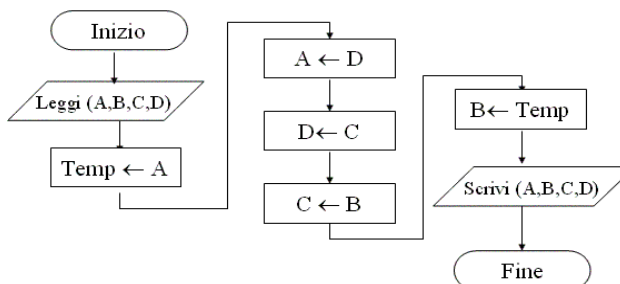
1) Tabelle delle variabili di input, di output e di lavoro

Nome	Descrizione	Tipo	Input	Output	Lavoro
A	Primo valore	Numero intero	X	X	
B	Secondo valore	Numero intero	X	X	
C	Terzo valore	Numero intero	X	X	
D	Quarto Valore	Numero intero	X	X	
TEMP	Variabile temporanea di appoggio	Numero intero			X

2) Algoritmo per passi

1. Inizio
2. Dammi i valori di A,B,C,D
3. Assegna a TEMP il valore di A
4. Assegna ad A il valore di D
5. Assegna ad D il valore di C
6. Assegna ad C il valore di B
7. Assegna ad B il valore di TEMP
8. Stampa i nuovi valori di A,B,C,D
9. Fine

3) Algoritmo col diagramma a blocchi



<pre>#include <iostream> using namespace std; int main() { int a,b,c,d,temp; cout<<"ROTAZIONE DI 1 STEP A DX\n"; cout<<"A= ";cin>>a; cout<<"B= ";cin>>b; cout<<"C= ";cin>>c; cout<<"D= ";cin>>d; temp=a; a=d; d=c; c=b; b=temp; cout<<"A="<<a<<"\n"; cout<<"B="<<b<<"\n"; cout<<"C="<<c<<"\n"; cout<<"D="<<d<<"\n"; return 0; }</pre>	<p>Dopo aver salvato il file e dopo averlo compilato se il programma mostra 0 errori e 0 warnings Allora è possibile eseguirlo.</p> <p>Test del Programma</p> <pre>ROTAZIONE DI 1 STEP A DX A= 1 B= 2 C= 3 D= 4 A=4 B=1 C=2 D=3</pre>
---	--

Problema4: Trovare l’algoritmo che calcoli la tabellina di un numero num scelto dall’utente da 1 a 1000 e stampi sul monitor del computer, la tabellina scelta

Per la realizzazione di un programma in C seguire i seguenti 5 passi:

1) Tabelle delle variabili di input, di output e di lavoro

Nome	Descrizione	Tipo	Input	Output	Lavoro
NUM	Numero da 1 a 1000 di cui si vuole conoscere la tabellina	Numero intero	X		
R1,...,R10	1°..1risultato della tabellina	Numero intero		X	

2) Algoritmo per passi

<ol style="list-style-type: none"> Inizio Dammi il numero, NUM Assegna a R1 il valore NUM*1 Assegna a R2 il valore NUM*2 Assegna a R3 il valore NUM*3 Assegna a R4 il valore NUM*4 Assegna a R5 il valore NUM*5 	<ol style="list-style-type: none"> Assegna a R6 il valore NUM*6 Assegna a R7 il valore NUM*7 Assegna a R8 il valore NUM*8 Assegna a R9 il valore NUM*9 Assegna a R10 il valore NUM*10 Stampa R1,R2,R3,R4,R5, R6,R7,R8,R9,R10 fine
--	--

<pre>#include <iostream> using namespace std; int main() { int num,r1,r2,r3,r4,r5,r6,r7,r8,r9,r10; cout<<"TABELLINA DEL ";cin>>num; r1=num*1; r2=num*2; r3=num*3; r4=num*4; r5=num*5; r6=num*6; r7=num*7; r8=num*8; r9=num*9; r10=num*10; cout<<r1<<" "; cout<<r2<<" "; cout<<r3<<" "; cout<<r4<<" "; cout<<r5<<" "; cout<<r6<<" "; cout<<r7<<" "; cout<<r8<<" "; cout<<r9<<" "; cout<<r10<<" "; return 0; }</pre>	<p>Test sul Programma</p> <pre>TABELLINA DEL 4 4 8 12 16 20 24 28 32 36 40 ----- Process exited after 3.081 seconds Premere un tasto per continuare . .</pre>
--	--

Problema6: Scrivere un algoritmo che dati tre voti interi A, B e C letti in input, calcoli la media aritmetica e la stampi in output

Per la realizzazione di un programma in C/C++ seguire i seguenti 5 passi:

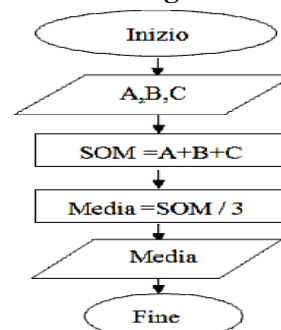
1) Tabelle delle variabili di input, di output e di lavoro					
Nome	Descrizione	Tipo	Input	Output	Lavoro
A	1° voto	Numero intero	X		
B	2° voto	Numero intero	X		
C	3° voto	Numero intero	X		
SOM	Somma dei tre voti	Numero intero			X
MED	Media dei tre voti	Numero decimale		X	

2) Algoritmo per passi

1. Inizio
2. Dammi il primo voto A
3. Dammi il secondo voto B
4. Dammi il terzo voto C
5. Somma i tre voti A, B e C
6. Calcola la media dei tre voti
7. Fine

```
#include <iostream>
using namespace std;
int main() {
    int a,b,c; float m;
    /* a,b,c sono i numeri, m è la loro media */
    cout<<"MEDIA ARITMETICA DI 3 NUM
\n";
    cout<<"A= "; cin>>a; cout<<"B= ";cin>>b;
    cout<<"C= "; cin>>c; m=(a+b+c)/3;
    cout<<"Media="<<m; return 0; }
```

3) Algoritmo col diagramma a blocchi



Test programma

```

MEDIA ARITMETICA DI 3 NUMERI
A= 2
B= 3
C= 4
Media=3
-----
Process exited after 6.231 seconds
Premere un tasto per continuare .
    
```

Strutture di selezione

1) Si parla di struttura di selezione ad una via, quando il percorso 'falso' non è presente. Essa, nel metodo del diagramma a blocchi, si rappresenta così:

<p>diagramma a blocchi</p>	<p>Esempio: N1 è positivo?</p>	<p>Sintassi in C++</p> <p>if (espressione) istruzione;</p> <p>Se ci sono più di una istruzione in una via, esse si racchiudono tra parentesi graffe</p> <p>if (espressione) { istruzione1; istruzione2; }</p>
-----------------------------------	---------------------------------------	--

2) Si parla di struttura di selezione a 2 vie, quando vi sono entrambi i percorsi: vero e falso. Essa, nel metodo del diagramma a blocchi, si rappresenta così:

<p>Nel diagramma a blocchi</p>	<p>Esempio: Max tra N1 e N2</p>	<p>Sintassi in C++</p> <pre> if (condizione) istruzione1; else { istruzione2; istruzione3; } </pre>
---------------------------------------	--	---

3) Si parla di **struttura di selezione a più vie**, quando oltre ai percorsi ‘vero’ e ‘falso’ ce ne sono altri (fig. a) Ad esempio: letto un prezzo di un prodotto e la sua quantità, in base alla quantità richiesta effettua uno sconto diverso. Essa, nel metodo del diagramma a blocchi, si rappresenta così (fig. b) e nel codice C++ si scrive usando il costrutto Switch-Case:

<p>Selezione Multipla (fig. a)</p>	<p>(fig. b)</p>	<p>Sintassi Switch-Case</p> <pre> switch (variabile) { case valore1: // tasso=1; break; case valore2: // tasso=2; break; default: /* ultimo caso */ // tasso=4; break; } </pre>
---	------------------------	--

4) si parla di **struttura di selezione annidata**, quando dentro una struttura di selezione ad una o più vie né è presenta almeno un'altra

<p>Selezione annidata</p>	<p>Sintassi in C++</p> <pre> if (condizione1) { /* se condizione è Vera */ else if (condizione2) istruzione a; else istruzione c; } else /* else relativo all'if piu esterno */ istruzione b; </pre>
----------------------------------	---

Esempi di problemi e programmi con le strutture di selezione annidata e multipla

Problema7: Letto un voto di un esame universitario (espresso in trentesimi 18-30), associare al voto una lettera A,B,C,D se rispettivamente il voto è > 28; >25, >22; >18.

<ol style="list-style-type: none"> 1. Inizio 2. Leggi voto 3. Se voto >28 allora 4. stampa (A) 5. Altrimenti se voto >25 6. Allora stampa (B) 7. Altrimenti se voto >22 8. Allora stampa (C) 9. Altrimenti se voto > 18 10. Allora stampa (D) 11. Fine se 12. Fine se 13. Fine se 14. Fine se 15. FINE 	<pre>#include <iostream> using namespace std; int main() { int voto; cout<<"Voto="; cin>>voto; if(voto>28) cout<<"Voto=A"; else if (voto>25) cout<<"Voto=B"; else if (voto>22) cout<<"Voto=C"; else if (voto>18) cout<<"Voto=D"; return 0; }</pre>	<p>Test programma</p> <pre>Voto=25 Voto=C</pre> <hr/> <pre>Voto=22 Voto=D</pre>
---	--	---

Attenzione: I “fine se” nel codice C++, non ci sono poiché all’interno di ogni if c’è una sola istruzione e non almeno due, requisito necessario per aggiungere la coppia di parentesi { }

Problema 8. Letto un prezzo di un prodotto e la sua quantità, in base alla quantità richiesta effettua uno tasso di sconto diverso (q=1,tasso=10%; q=2, tasso=20%; q=3, tasso=30%; altro, tasso= 40%.)

1) Tabelle delle variabili di input, di output e di lavoro					
Nome	Descrizione	Tipo	Input	Output	Lavoro
prezzo	Prezzo del prodotto	decimale	X		
quantità	Quantità dello stesso prodotto	intero	X		
ilordo	Importo lordo	decimale			X
inetto	Importo netto da pagare	decimale		X	
sconto	Sconto da applicare all’importo lordo	intero			X
tasso	Tasso di sconto in base alla quantità	intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. Inizio 2. Leggi prezzo (del prodotto) 3. Leggi quantità (del prodotto) 4. Calcola ilordo = prezzo*quantità scegli: 5. se la quantità = 1 allora tasso = 10 6. Se la quantità = 2 allora tasso = 20 7. Se la quantità = 3 allora tasso = 30 8. Negli altri casi, altrimenti tasso = 40 9. Calcola sconto = ilordo*tasso /100 10. Calcola inetto = ilordo - sconto 11. Stampa a video l’importo netto 12. Fine 	<p>3) Codice del Programma</p> <pre>int main() { int tasso,q,sconto; float prezzo, ilordo, inetto; cout<<"Prezzo=";cin>>prezzo; cout<<"Quantita=";cin>>q; ilordo=prezzo*q; switch (q) { case 1: tasso=10; break; case 2: tasso=20; break; case 3: tasso=30; break; default: tasso=40; break; } sconto=ilordo*tasso/100; inetto=ilordo-sconto; cout<<"inetto="<<inetto; return 0;}</pre>	<p>Test programma</p> <pre>Prezzo=100 Quantita=1 inetto=90</pre> <hr/> <pre>Prezzo=10 Quantita=2 inetto=16</pre>
---	--	--

Dati due numeri N1 e N2, determinare il minimo e il massimo e stamparli

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Numero intero	X		
N2	Secondo Numero	Numero intero	X		
MIN	Minimo tra i due numeri	Numero intero		X	
MAX	Massimo tra due numeri	Numero intero		X	

<pre>#include <iostream> using namespace std; int main() { int n1,n2,max,min; /* n1, n2 sono i due numeri; max e min sono il massimo e il minimo */ cout<<"max e min di 2 numeri \n"; cout<<"N1= ";cin>>n1;</pre>	<pre>cout<<"N2= ";cin>>n2; if(n1>n2) { max=n1; min=n2; } else { max=n2; min=n1; } cout<<"Max="<<max; cout<<" Min="<<min; return 0; }</pre>	<div style="border: 1px solid black; padding: 5px;"> <p>Test del programma</p> <pre>MAX E MIN DI DUE NUMERI N1= 4 N2= 2 Max=4 Min=2</pre> </div>
---	---	---

Problema9: Dati due numeri N1 e N2, ordinarli in ordine crescente e stamparli a video

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Numero intero	X		
N2	Secondo Numero	Numero intero	X		

<p>2) Programma</p> <pre>#include <iostream> using namespace std; int main() { int n1,n2; cout<<"Ordine crescente di 2 numeri \n"; cout<<"N1= ";cin>>n1; cout<<"N2= ";cin>>n2; if(n1>n2) { cout<<n2<<" "; cout<<n1; } else { cout<<n1<<" "; cout<<n2; } return 0; }</pre>	<p>3) Algoritmo col diagramma a blocchi Non disegnato per mancanza di tempo</p> <div style="text-align: center;"> <p>Test del programma</p> <pre>Ordine Crescente DI 2 Numeri N1= 3 N2= 1 1 3 -----</pre> </div>
---	--

Problema10: Dati due numeri N1, N2, tale che N1*N2<0, calcolare quello positivo e quello negativo e stamparlo

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Numero intero	X		
N2	Secondo Numero	Numero intero	X		
POS	Numero Positivo	Numero Intero		X	
NEG	Numero Negativo	Numero Intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. Leggi N1 e N2 tale che $N1 * N2 < 0$ 3. se $N1 > 0$, allora POS=N1 altrimenti NEG=N1 4. se $N2 > 0$, allora POS=N2 altrimenti NEG=N2 5. stampa POS, NEG 6. fine 	<p style="text-align: center;">Test programma</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>Positivo e Negativo DI 2 Numeri N1= 3 N2= -2 il positivo e' 3 il negativo e' -2</pre> </div>
--	--

Problema11: Dato un numero Num, determinare se pari o dispari (uso funzione % del C++)

1) Tabelle delle variabili di input e output							
Nome	Descrizione	Tipo	Input	Output	Lavoro		
Num	Numero	intero	X				
<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi un numero Num 3. se la parte intera di $Num/2 = 0$ 4. allora stampa N pari 6. altrimenti stampa N dispari 7. fine 			<p>3) Algoritmo col diagramma a blocchi</p> <pre>int main() { int num; cout<<"num=";<<cin>>num; if (num%2 == 0) { cout<<num; cout<<" e' un numero pari"; } else { cout<<num; cout<<" e' un numero dispari"; } return 0; }</pre>				
<p>Test sul programma in C++</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px; width: 50%; font-family: monospace;"> num=3 3 e' un numero dispari </td> <td style="border: 1px solid black; padding: 2px; width: 50%; font-family: monospace;"> num=4 4 e' un numero pari </td> </tr> </table>						num=3 3 e' un numero dispari	num=4 4 e' un numero pari
num=3 3 e' un numero dispari	num=4 4 e' un numero pari						

Problema12: Dati tre numeri N1,N2 e N3 determinare quello Max e stamparlo a video

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Intero	X		
N2	Secondo Numero	Intero	X		
N3	Terzo Numero	Intero	X		
MAX	Massimo da calcolare	Intero		X	
<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi i valori N1, N2, N3 3. Assegna a Max, il numero N1 4. Se $N2 > Max$ allora a Max assegna N2 5. Se $N3 > Max$ allora a Max assegna N3 6. stampa Max 7. fine 			<p>3) Programma in C++</p> <pre>int main() { int n1,n2,n3,max; cout<<"N1=";<<cin>>n1; cout<<"N2=";<<cin>>n2; cout<<"N3=";<<cin>>n3; max=n1; if (n2>max) max=n2; if (n3>max) max=n3; cout<<"Il valore massimo e' "<<max; return 0; }</pre>		
<p>Test del programma</p> <pre>N1=1 N2=4 N3=2 Il valore massimo e' 4</pre>					

PROBLEMA13: Scrivere un algoritmo che dati 3 numeri N1, N2 e N3 positivi e negativi, calcoli la somma solo dei numeri positivi e la stampi a video.

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Intero	X		
N2	Secondo Numero	Intero	X		
N3	Terzo Numero	Intero	X		
SOM	<u>Somma dei solo numeri positivi</u>	Intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi i valori N1, N2, N3 3. Se N1 > 0 allora a SOM=SOM+N1 4. Se N2 > 0 allora a SOM=SOM+N2 5. Se N3 > 0 allora a SOM=SOM+N3 6. stampa SOM 7. fine <p>Test del programma</p> <pre>N1=2 N2=-3 N3=5 La somma e' ?</pre>	<p>3) #include <iostream></p> <pre>using namespace std; int main() { int n1,n2,n3,som; cout<<"N1=";<<cin>>n1; cout<<"N2=";<<cin>>n2; cout<<"N3=";<<cin>>n3; if (n1>0) som=som+n1; if (n2>0) som=som+n2; if (n3>0) som=som+n3; cout<<"La somma e' "<<som; return 0; }</pre>
--	--

La variabile conteggio

Una variabile conteggio è un contatore che viene inizializzato al valore zero¹ e poi viene incrementato di una unità ogni qual volta si verifica una certa condizione.

PROBLEMA14: Scrivere un algoritmo che dati 3 numeri N1, N2 e N3 positivi e negativi, conti quanti sono positivi e li stampi a video. (usare una variabile conteggio contapos).

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	intero	X		
N2	Secondo Numero	intero	X		
N3	Terzo Numero	intero	X		
CPOS	<u>Contatore di numeri positivi</u>	intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi i valori N1, N2, N3 3. Se N1 > 0 allora CPOS=CPOS+1 4. Se N2 > 0 allora CPOS=CPOS+1 5. Se N3 > 0 allora CPOS=CPOS+1 6. stampa CPOS 7. fine <p>Test del Programma</p> <pre>N1=1 N2=2 N3=-5 Il numero dei positivi e' 2</pre>	<p>3) Programma in C++</p> <pre>int main() { int n1,n2,n3,cpos; cout<<"N1=";<<cin>>n1; cout<<"N2=";<<cin>>n2; cout<<"N3=";<<cin>>n3; if (n1>0) cpos++; if (n2>0) cpos++; if (n3>0) cpos++; cout<<"Il numero dei positivi e' "<<cpos; return 0; }</pre>
---	--

¹ Per far capire il perché la variabile conteggio viene inizializzata a zero, dire che essa è simile al contatore del contachilometri dell'automobile o al contasecondi del cronometro di un orologio: tutti partono da zero.

In seguito, però, ci saranno anche variabili a conteggio che saranno inizializzate ad un valore diverso come ad es. la variabile indice ciclo (ad 1, a 2) ecc.

PROBLEMA15: Scrivere un algoritmo che dati 3 numeri N1, N2 e N3 pari e dispari, conti quanti sono pari e li stampi a video. (Suggerimento: usare una variabile conteggio cpari)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
N1	Primo Numero	Numero intero	X		
N2	Secondo Numero	Numero intero	X		
N3	Terzo Numero	Numero intero	X		
CPARI	Contatore dei numeri pari	Numero intero		X	

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi i valori N1, N2, N3 5. Se $N1\%2 = 0$ allora $CPARI+=1$ 6. Se $N2\%2 = 0$ allora $CPARI+=1$ 7. Se $N3\%2 = 0$ allora $CPARI+=1$ 8. stampa CPARI 9. fine <p>Test del Programma</p> <pre>N1=1 N2=2 N3=3 Il numero dei pari e' 1</pre>	<p>3) Algoritmo col diagramma a blocchi</p> <pre>int main() { int n1,n2,n3,cpari; cout<<"N1=";cin>>n1; cout<<"N2=";cin>>n2; cout<<"N3=";cin>>n3; if (n1%2==0) cpari++; if (n2%2==0) cpari++; if (n3%2==0) cpari++; cout<<"Il numero dei pari e' "<<cpari; return 0; }</pre>
--	--

PROBLEMA16: Scrivere un algoritmo che dati 3 numeri a, b e c li ordini in modo strettamente crescente. (Suggerimento: basta calcolare prima il minimo tra a e b ed tra a e c; poi calcolare il minimo tra b e c; usare una variabile scambio)

1) Tabelle delle variabili di input e output					
Nome	Descrizione	Tipo	Input	Output	Lavoro
a	Primo Numero	intero	X	X	
b	Secondo Numero	intero	X	X	
c	Terzo Numero	intero	X	X	
T	Variabile Temporanea	intero			X

<p>2) Algoritmo per passi</p> <ol style="list-style-type: none"> 1. inizio 2. dammi i valori a, b, c 3. se $a > b$ scambiali di posto in modo che il minimo sia al 1) posto 4. se $a > c$ scambiali di posto in modo che il minimo sia al 1) posto 5. se $b > c$ scambiali di posto in modo che il minimo sia al 2) posto 6. stampa a video i nuovi valori di a,b e c 7. fine <p>Test sul Programma</p> <pre>Ordinamento cresc. di 3 n.ri A=3 B=2 C=1 I numeri sono 1 2 3</pre>	<p>3) Codifica programma C++</p> <pre>#include <iostream> using namespace std; int main() { int a,b,c,t; cout<<"Ordin. cresc. di 3 n.ri \n"; cout<<"A=";cin>>a; cout<<"B=";cin>>b; cout<<"C=";cin>>c; if (a>b) {t=a; a=b; b=t; } if (a>c) {t=a; a=c; c=t; } if (b>c) {t=b; b=c; c=t; } cout<<"I numeri: "<<a<<" "; cout<<b<<" "; cout<<c; return 0; }</pre>
--	---

Problema17: Scrivere un algoritmo che dato il prezzo di uno smartphone LG K10 2017 e data la quantità dei pezzi venduti, calcoli sull'importo totale: lo sconto del 10% se la quantità dei pezzi venduti è da 1 a 5; lo sconto del 20% se la quantità dei pezzi venduti è da 6 a 10; lo sconto del 30% se la quantità dei pezzi venduti è da 11 a 15; lo sconto del 40% se il n.ro di pezzi venduti è > 15.

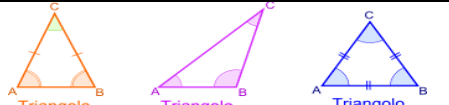
Calcolare, oltre all'importo totale al netto dell'iva, l'iva del 22%, l'importo totale con l'iva, il tasso di sconto e infine l'importo totale scontato.

Suggerimento per controllare tutti e quattro i casi si può usare a scelta del programmatore o la struttura di selezione annidata if-else-if-else oppure la struttura switch. Entrambe le strutture permettono di eseguire solo la via associata all'intervallo dei pezzi inseriti, escludendo le altre vie, il che permette una complessità computazionale ottimale.

1) Tabelle delle variabili e costanti					
Nome	Descrizione	Tipo	Input	Output	Lavoro
Tasso	Tasso di sconto in base ai pezzi	Intero	X		
Pezzi	N.ro di pezzi dello stesso smartphone	intero	X		
Costo	Costo di un singolo smartphone	decimale			X
itnetto	Importo totale al netto dell'iva	decimale		X	
iva	Iva del 22% sull'importo totale	decimale		X	
totiva	Importo totale ivato	decimale		X	
sconto	Sconto da applicare al totale	decimale		X	
pscontato	Prezzo scontato	decimale		X	

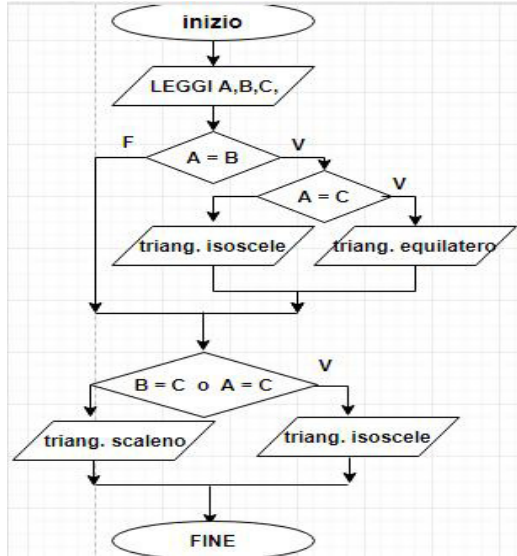
<p>2) Algoritmo con la struttura switch</p> <ol style="list-style-type: none"> Inizio Leggi il costo di uno smartphone LG k10 Leggi il n.ro di pezzi di quel smartphone Calcola itnetto = costo*n.ro dei pezzi Calcola iva=itnetto*22/100 Calcola il tot. iva=itnetto+iva Se i pezzi sono fino a 5 allora tasso = 10 altrimenti se pezzi sono (6 -10) → tasso = 20 altrimenti se pezzi sono (11-15) → tasso = 30 altrimenti applico il tasso = 40 Calcolo sconto = tot.iva*tasso/100 Calcolo pscontato = tot.iva – sconto Stampa del pscontato Fine <pre> totale stock di smartphone LG k10 Costo=100 Pezzi=2 inetto=200 iva=44 totiva=244 sconto=24.4 Il totale da pagare e' 219.6 </pre>	<p>3) Programma in C++</p> <pre> int main() { int pezzi, tasso; float iva, costo, itnetto, totiva, sconto, pscontato; cout<<"Totale stock di smartphone LG k10 2017\n"; cout<<"Costo="; cin>>costo; cout<<"Pezzi="; cin>>pezzi; itnetto=pezzi*costo; iva=itnetto*22/100; totiva=itnetto+iva; switch (pezzi) { case 1 ... 5: tasso=10; break; case 6 ... 10: tasso=20; break; case 11 ... 15: tasso=30; break; default: tasso=40; break; } sconto=totiva*tasso/100; pscontato=totiva-sconto; cout<<"inetto="<<itnetto; cout<<" iva="<<iva; cout<<" totiva="<<totiva; cout<<" sconto="<<sconto<<endl; cout<<"Il totale da pagare e' "<<pscontato; return 0; } </pre>
---	---

Problema18: Scrivere un programma che legga da input le lunghezze dei lati di un triangolo e determini se il triangolo è equilatero, isoscele o scaleno, stampando l'opportuno messaggio.

	<p>Analisi del problema. Un triangolo è isoscele se due lati qualsiasi sono uguali. Scaleno se tutti e tre i lati sono diversi. Equilatero se tutti e tre sono uguali.</p>
---	---

Algoritmo – con la struttura if –elseif-else

Dal sito internet: <https://www.draw.io/>



Programma in C++

```
int main() {
    int a, b, c;
    cout<<"Inserisci i 3 lati del triangolo"<<endl;
    cout<<"a= "; cin>>a;
    cout<<"b= "; cin>>b;
    cout<<"c= "; cin>>c;
    if (a == b)
    {
        if (a == c)
            cout<<"triangolo equilatero";
        else /* solo a<>c ma è sempre a=c */
            cout<<"triangolo isoscele";
    } /* cioè a<>b */
    else if (b == c || a == c)
        cout<<"triangolo isoscele";
    else /* anche b<>c e a<>c, oltre ad a<>b */
        cout<<"triangolo scaleno";
    return 0; }
```

Inserisci i 3 lati di un triangolo
a= 4
b= 4
c= 4
triangolo equilatero

Inserisci i 3 lati di un triangolo
a= 1
b= 2
c= 3
triangolo scaleno

Problema19. Realizzare un programma che, presi in input 2 numeri N1 e N2 reali e un operatore (+, -, *, /), esegue l'operazione stampandone il risultato. Nel caso l'operatore inserito sia diverso dai quattro operatori, si stampi il messaggio non eseguibile.

1. Inizio
2. Leggi N1, N2, Operatore
3. Scegli Operatore
Inizio switch
case '+': stampa N1+N2; break;
case '-': stampa N1-N2; break;
case '*': stampa N1*N2; break;
case '/': stampa N1/N2; break;
default: "operazione non eseguibile"; break;
4. Fine switch
5. Fine

```
int main() {
    int N1, N2; char operatore;
    cout<<"Operazione (+,-,*) tra N1 e N2\n";
    cout<<"N1=";cin>>N1; cout<<"N2=";cin>>N2;
    cout<<"Operatore=";cin>>operatore;
    switch (operatore) {
        case '+': cout<<"N1+N2= "<<N1+N2; break;
        case '-': cout<<"N1-N2= "<<N1-N2; break;
        case '/': cout<<"N1/N2= "<<N1/N2; break;
        case '*': cout<<"N1*N2= "<<N1*N2; break;
        default: cout<<"Non eseguibile"; break;
    } return 0; }
```

Test del Programma

Calcolo operazione (+,-,*) tra due Numeri
N1=2
N2=3
Operatore=+
N1+N2= 5

Calcolo operazione (+,-,*) tra due Numeri
N1=2
N2=5
Operatore=*
N1*N2= 10

Calcolo operazione (+,-,*) tra due Numeri
N1=10
N2=3
Operatore=-
N1-N2= 7

Problema20. Calcolare la radice x di una equazione di 1 grado $ax+b=0$;

1. Inizio
2. leggi a, b
3. se $a < 0$ allora $x = -b/a$, scrivi x
4. altrimenti scrivi "eq. impossibile", fine se
5. Fine

```
int main(){
    int a, b; float x;
    cout<<"Coefficiente della x: ";cin>>a;
    cout<<"Termine noto: ";cin>>b;
    if (a!=0)
        { x=(-b/a); cout<<"x= "<<x; }
    else cout<<"Equazione impossibile";
    return 0; }
```

Soluzione dell' Eq. $ax+b=0$
Coefficiente della x: 2
Termine noto: -2
x= 1

Soluzione dell' Eq. $ax+b=0$
Coefficiente della x: 0
Termine noto: 5
Equazione impossibile

Problema 21. Calcolare le radici x1 e x2 di una equazione di 2 grado con $\Delta \geq 0$

<p>Ricordiamo che:</p> <p><i>equazione completa</i> $ax^2 + bx + c = 0$</p> <p>$\Delta > 0$ — $x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$</p> <p>$\Delta < 0$ — $x_1 = x_2 = -\frac{b}{2a}$</p> <p>$\Delta = 0$ — <i>equazione impossibile</i></p> <p>Algoritmo</p> <ol style="list-style-type: none"> Inizio leggi a,b,c calcola $\Delta = b^2 - 4ac$ calcola $x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ calcola $x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ se delta = 0 allora soluzioni coincidenti altrimenti se delta < 0 allora "non ci sono soluzioni reali altrimenti "le soluzioni sono:" x1 e x2 fine 		<pre>#include <iostream> #include <math.h> /* per usare la funzione matematica radice sqrt*/ int main() { using namespace std; double a,b,c,x1,x2,delta; cout<<"Calcolo radici di una eq. di 2 grado"<<endl; cout<<"Coefficiente della x^2: ";cin>>a; cout<<"Coefficiente della x: ";cin>>b; cout<<"Termine noto: ";cin>>c; delta = ((b*b)-(4*a*c)); cout<<"il delta e' "<<delta<<endl; if (delta > 0) { x1 = (-b)-(sqrt(delta))/(2*a); x2 = (-b)+(sqrt(delta))/(2*a); cout<<"x1: "<<x1<<" x2: "<<x2<<endl; } else if (delta < 0) cout<<"Non ci sono soluzioni reali"<<endl; else /* soluzioni reali e coincidenti */ cout<<" x1=x2: "<<((-b)/(2*a)); return 0; }</pre>
<p>Calcolo radici di una eq. di 2 grado Coefficiente della x^2: 2 Coefficiente della x: 3 Termine noto: -5 il delta e' 49 Le soluzioni sono: x1: -2.5 x2: 1</p>	<p>Calcolo radici di una eq. di 2 grado Coefficiente della x^2: 1 Coefficiente della x: -2 Termine noto: 1 il delta e' 0 Le soluzioni sono coincidenti: x1: 1 x2: 1</p>	<p>Calcolo radici di una eq. di 2 grado Coefficiente della x^2: 2 Coefficiente della x: -2 Termine noto: 1 il delta e' -4 Non ci sono soluzioni reali</p>

Problema 22. Scrivere un algoritmo che data una temperatura temp, stampa a video il messaggio:
a) "caldo africano" se la temperatura è compresa tra 41 e 50; b) " molto caldo " se la temperatura è compresa tra 33 e 40; c) "caldo" se la temperatura è compresa tra 25 e 32; d) " mite " se la temperatura è compresa tra 17 e 24; e) " fresco " se la temperatura tra 9 e 16; f) " freddo " se la temperatura è tra 1 e 8 ; g) " molto freddo " se la temperatura è altrimenti.

<p>Algoritmo</p> <ol style="list-style-type: none"> Inizio Leggi temp; switch (temp) <ul style="list-style-type: none"> Inizio case case 41...50; scrivi "caldo africano" case 33...40; scrivi "molto caldo" case 25...32; scrivi "caldo" case 17...24; scrivi "mite" case 9...16; scrivi "fresco" case 1...8; scrivi "freddo" default: scrivi "molto freddo" Fine case Fine 	<pre>#include <iostream> using namespace std; int main() { int temp; cout<<"Temperatura = ";cin>>temp; switch (temp) { case 41 ... 50: cout<<"caldo africano"; break; case 33 ... 40: cout<<"molto caldo"; break; case 25 ... 32: cout<<"caldo"; break; case 17 ... 24: cout<<"mite"; break; case 9 ... 16: cout<<"fresco"; break; case 1 ... 8: cout<<"freddo"; break; default: cout<<"molto freddo"; break; } return 0; }</pre>	
<p>Temperatura = 45 caldo africano ----- Process exited with return value 0 Press any key to continue . . . =</p>	<p>Temperatura = 30 caldo ----- Process exited with return value 0 Press any key to continue . . .</p>	<p>Temperatura = -2 molto freddo ----- Process exited with return value 0 Press any key to continue . . .</p>