



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Informatica

Modulo di Programmazione

INFORMATICA
MODULO DI PROGRAMMAZIONE
C++ BASICS

mauro.pelucchi@gmail.com

Mauro Pelucchi

2023/2024

Agenda

- Operators and Expressions
- Basic I/O
- Control-Flow Constructs (while, for)

Common Input/output functions

- cin
 - Standard input stream
 - Normally keyboard
- cout
 - Standard output stream
 - Normally computer screen
- cerr
 - Standard error stream
 - Display error messages

Preprocessor directives

- Processed by preprocessor before compiling
- Begin with #

```
#include <iostream>  
#include <iomanip> // to import setfill(...), setw(...), setprecision(...)
```

Printing a Line of Text

- Standard output stream object
 - `std::cout`
 - `<<` `>>` Stream insertion operator
 - Value to right (right operand) inserted into output stream
- Input stream object
 - `>>` `<<` Stream extraction operator
 - Used with `std::cin`
 - Waits for user to input value, then press Enter (Return) key
 - Stores value in variable to right of operator
 - Converts value to variable data type
- Namespace
 - `std::` specifies using name that belongs to “namespace” `std`
 - `std::` removed through use of using statements
- Escape characters
 - `\` `>>` Indicates “special” character output

Escape sequence

Escape Sequence	Description
<code>\n</code>	Newline. Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line.
<code>\a</code>	Alert. Sound the system bell.
<code>\\</code>	Backslash. Used to print a backslash character.
<code>\"</code>	Double quote. Used to print a double quote character.

Variables

- Location in memory where value can be stored
- Common data types
 - int - integer numbers
 - char - characters
 - double - floating point numbers
- Declare variables with name and data type before use
- Can declare several variables of same type in one declaration
 - Comma-separated list
 - `int integer1, integer2, sum;`

Esercizio 1

Può un neonato vivere per 1 miliardo di secondi?

Scrivere un programma C++ in grado di visualizzare il numero di anni compiuti da un neonato dopo 1 miliardo di secondi, facendo uso di variabili di tipo **double**.

Nota: il seguente statement C++:
double secondi = 1e+9;

dichiara e assegna alla variabile **secondi** il valore **1 miliardo**, in notazione scientifica.


```
#include <iostream>
#include <iomanip> // to import setfill(...), setw(...), setprecision(...)

using namespace std;

int main() {

    double seconds = 1e+9;
    double hours, days, years;

    hours = seconds / 3600.0;
    days = hours / 24.0;
    years = days / 365.0;

    cout << "\n A child, after " << seconds << " seconds, is ";
    cout << setfill('.') << setw(5) << setprecision(4) << years;
    cout << "   years old! " << endl << endl;

    // wait for user to hit enter or another key
    std::cin.sync();
    std::cout << "press enter to exit...";
    std::cin.ignore();
    //
    // for Windows Users use
    //
    // system("pause");
    return 0;
} // end main
```

exercise_1.cpp

Esercizio 2

Acquisire da tastiera le coordinate di tre punti nel piano cartesiano, riconoscere se possono rappresentare i vertici di un triangolo non degenere e, in caso affermativo, mostrarne a video il perimetro indicando anche se il triangolo è: equilatero, isoscele, scaleno o rettangolo.

```

#include <iostream>
#include <cmath>      // to import sqrt(...), abs(...)
#include <iomanip>

using namespace std;
int main() {

    float ax,ay;
    float bx,by;
    float cx,cy;

    float lab, lac, lbc;
    float perimeter;

    cout << "\n Digit ax: "; cin >> ax;
    cout << "\n Digit ay: "; cin >> ay;

    cout << "\n Digit bx: "; cin >> bx;
    cout << "\n Digit by: "; cin >> by;

    cout << "\n Digit cx: "; cin >> cx;
    cout << "\n Digit cy: "; cin >> cy;

    lab = sqrt((ax-bx)*(ax-bx)+(ay-by)*(ay-by));
    lac = sqrt((ax-cx)*(ax-cx)+(ay-cy)*(ay-cy));
    lbc = sqrt((bx-cx)*(bx-cx)+(by-cy)*(by-cy));

    /*
    the right test is not x == y
    but (x-y < 1e-12) {...}

    check always the precision of match functions and your variables
    */
}

```

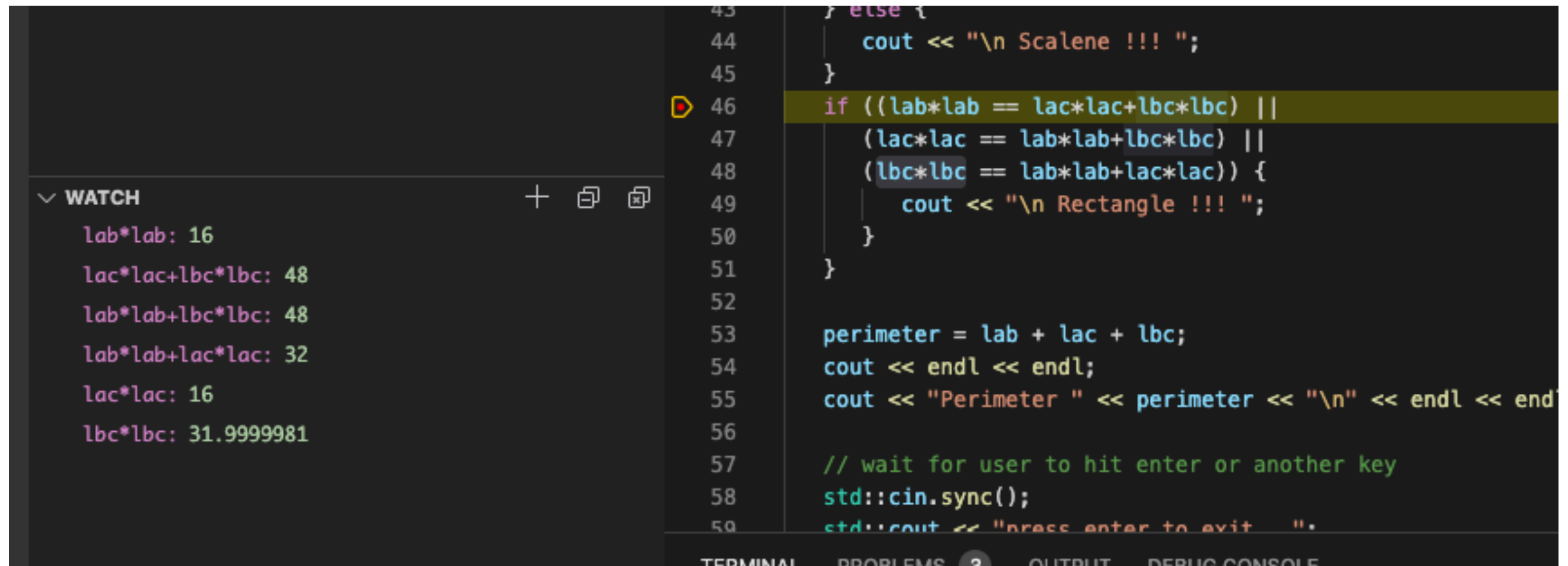
exercise_2.cpp

```
if ( !(lab < lac+lbc && lac < lab+lbc && lbc < lab+lac) ) {
    cout << "\n Degenerate triangle !!! ";
} else {
    if (lab == lac && lac == lbc) {
        cout << "\n Equilateral !!! ";
    } else if ((lab == lac) || (lab == lbc) || (lac == lbc)) {
        cout << "\n Isosceles !!! ";
    } else {
        cout << "\n Scalene !!! ";
    }
    if (((lab*lab)-(lac*lac+lbc*lbc) < 1e-12) ||
        ((lac*lac)-(lab*lab+lbc*lbc) < 1e-12) ||
        ((lbc*lbc)-(lab*lab+lac*lac) < 1e-12)) {
        cout << "\n Rectangle !!! ";
    }
}

perimeter = lab + lac + lbc;
cout << endl << endl;
cout << "Perimeter " << perimeter << "\n" << endl << endl;

// wait for user to hit enter or another key
std::cin.sync();
std::cout << "press enter to exit...";
std::cin.ignore();
//
// for Windows Users use
//
// system("pause");
return 0;
} // end main
```

Esercizio 2



The screenshot shows a C++ IDE with a watch window on the left and source code on the right. The watch window displays the following expressions and their values:

Expression	Value
lab*lab	16
lac*lac+lbc*lbc	48
lab*lab+lbc*lbc	48
lab*lab+lac*lac	32
lac*lac	16
lbc*lbc	31.9999981

The source code on the right is as follows:

```
43 } else {  
44     cout << "\n Scalene !!! ";  
45 }  
46 if ((lab*lab == lac*lac+lbc*lbc) ||  
47     (lac*lac == lab*lab+lbc*lbc) ||  
48     (lbc*lbc == lab*lab+lac*lac)) {  
49     cout << "\n Rectangle !!! ";  
50 }  
51 }  
52  
53 perimeter = lab + lac + lbc;  
54 cout << endl << endl;  
55 cout << "Perimeter " << perimeter << "\n" << endl << endl;  
56  
57 // wait for user to hit enter or another key  
58 std::cin.sync();  
59 std::cout << "press enter to exit ";
```

Esercizio 3

Scrivere un programma che acquisisca da tastiera una sequenza di caratteri terminata da !
e riporti sul monitor una sequenza derivata
dalla precedente secondo le regole dell'F-alfabeto
(ogni vocale viene raddoppiata inserendovi in mezzo una f).

Esempio:

quanto mi piace questo corso!

qufuafantofo mifi pifiafacefe qufuefestofco forsofo!

Per semplicità si considerino solo le lettere minuscole.

Ogni carattere diverso dalle vocali deve restare inalterato.

Si acquisisca da tastiera un carattere alla volta.

```
#include <iostream>
using namespace std;
```

```
int main( ) {
    char c;
    cout << "Digit a sentences: \n";
    do {
        cin.get(c);
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')
            cout << c << "f" << c;
        else
            cout << c;
    } while (c != '!');
    cout << endl;

    // wait for user to hit enter or another key
    std::cin.sync();
    std::cout << "press enter to exit...";
    std::cin.ignore();
    //
    // for Windows Users use
    //
    // system("pause");
    return 0;
} // end main
```

exercise_3.cpp

Esercizio 4

(soluzione omessa per esercizio individuale)

Uno studente deve sostenere una prova pratica e una prova teorica per superare un esame.

A ogni prova è assegnato un punteggio tra 0 e 30.

Il voto finale è dato dalla media dei due.

Anche se il voto finale è maggiore o uguale a 18, lo studente risulterà promosso solo se entrambe le prove sono sufficienti, altrimenti dovrà risostenere la prova che è risultata insufficiente.

Scrivere un programma che acquisisca i voti delle due prove e riporti a video un messaggio che descrive le sorti dello studente

Esercizio 5

(soluzione omessa per esercizio individuale)

Dato un numero intero tra 1 e 12, che rappresenta il mese corrente, stampare il nome del mese per esteso ("Gennaio" ... "Dicembre")

Esercizio 6

(soluzione omessa per esercizio individuale)

Si scriva un programma che implementi una semplice calcolatrice in grado di compiere le 4 operazioni (+ − × ÷) tra numeri interi.

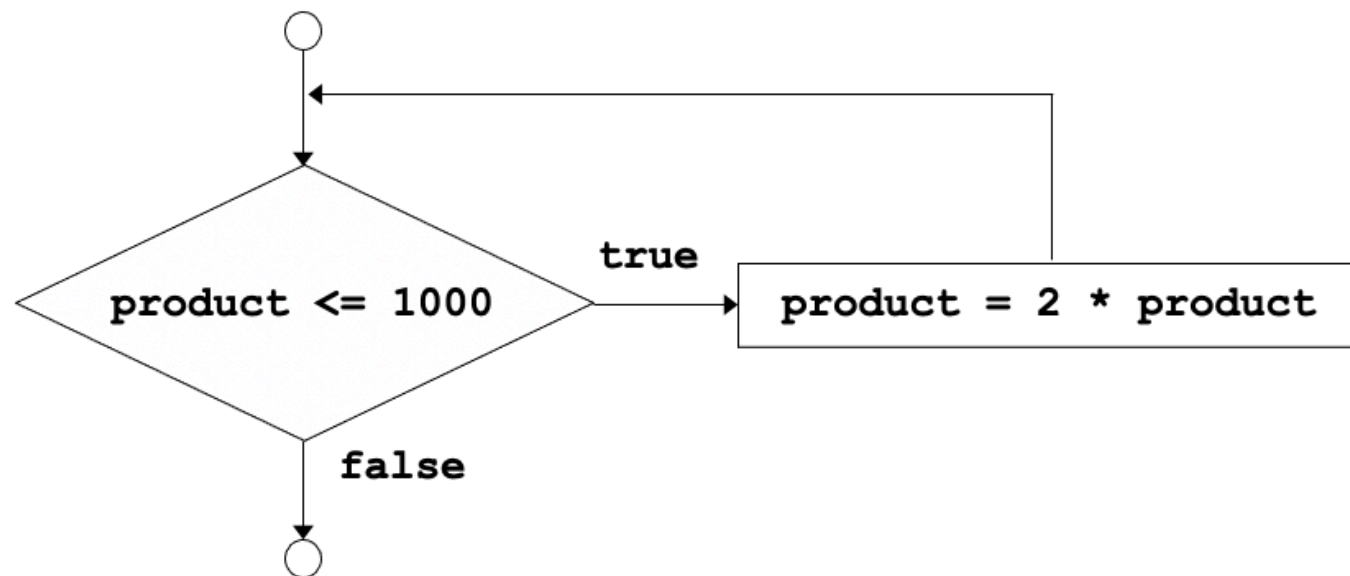
Il programma presenti un semplice menù da cui l'utente indichi (con un numero tra 1 e 4) l'operazione da svolgere.

In seguito il programma acquisirà da tastiera i due operandi e stamperà il risultato dell'operazione.

Control-Flow Constructs

while Repetition Structure

Action repeated while some condition remains true

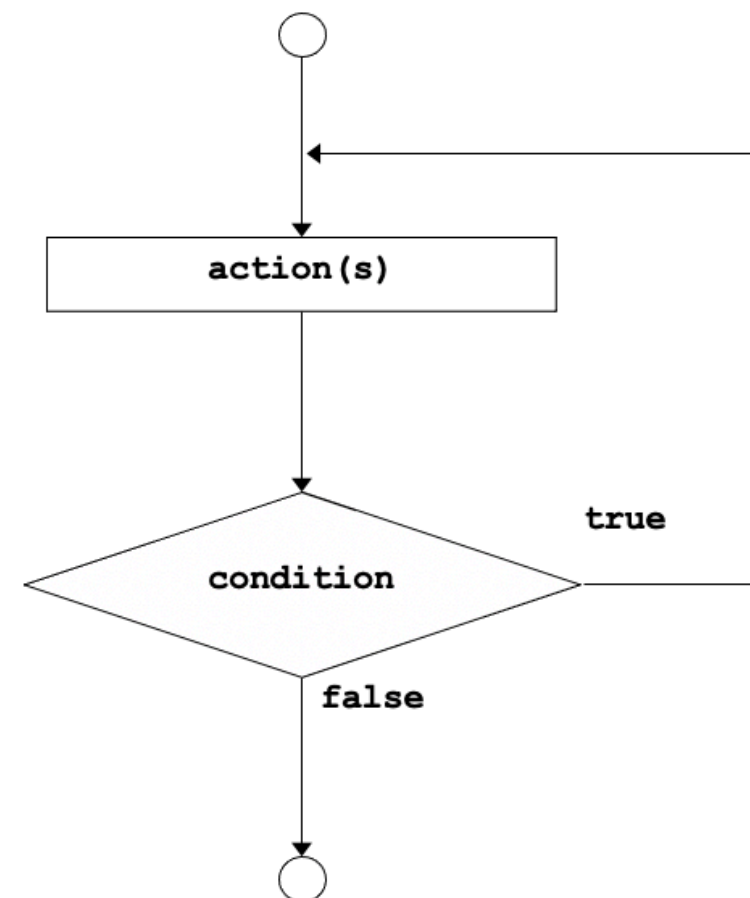


```
int product = 2;  
while ( product <= 1000 )  
    product = 2 * product;
```

do/while Repetition Structure

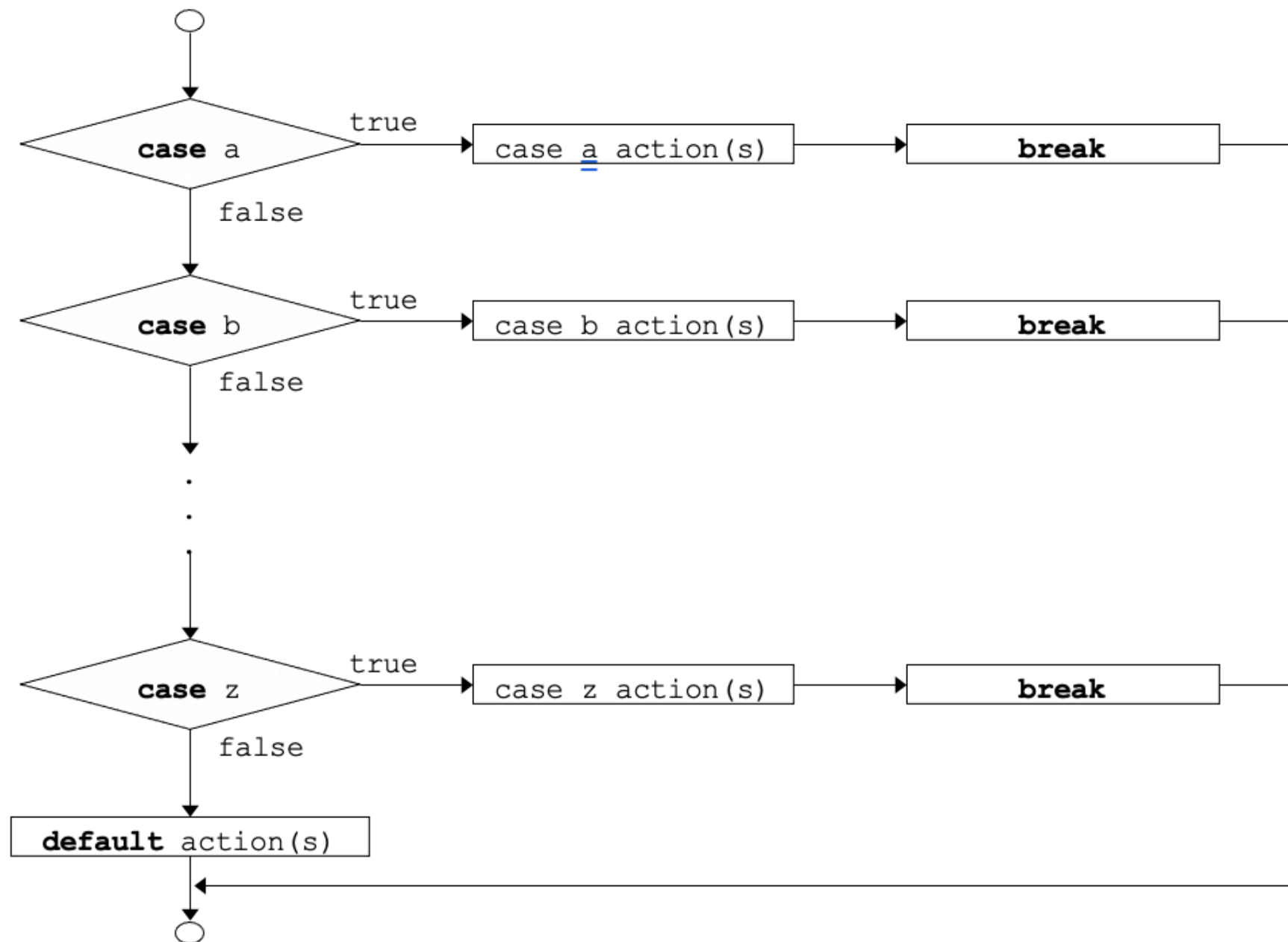
Makes loop continuation test at end, not beginning

```
do {  
    statement  
} while ( condition );
```



switch

Multiple-Selection Structure



break Statements

- break statement
 - Immediate exit from while, for, do/while, switch
 - Program continues with first statement after structure
 - Common uses: Escape early from a loop; Skip the remainder of switch

Esercizio 1

Dato un numero positivo Q ,

- scrivere la sua rappresentazione in binario naturale, applicando il tradizionale algoritmo per divisioni successive (per convenzione, in questo esercizio l'output si intende corretto se letto da destra a sinistra);
- indicare anche il minimo numero di bit utilizzato.

Input: 19 in decimale, Output: con 5 bit = 10011 in binario.

(MSB) (LSB)

Esercizio 2

Si scriva un programma in linguaggio C++ che letto un numero intero positivo dallo standard input, visualizzi a terminale il quadrato del numero stesso facendo uso soltanto di operazioni di somma.

Si osservi che il quadrato di ogni numero intero positivo N può essere costruito sommando tra loro i primi N numeri dispari.

Esempio:

$$N = 5;$$

$$N^2 = 1 + 3 + 5 + 7 + 9 = 25.$$

Esercizio 2

Algoritmo - Linguaggio Informale

L'idea è quella di scandire i primi N numeri dispari esprimendoli nella forma

$$1 + i + 1$$

al variare dell'intero i tra 0 e $N-1$ e accumulare la loro somma man mano che si procede nella loro scansione in un'altra variabile.

$$N^2 = (2 \cdot 0 + 1) + \dots + (2 \cdot i + 1) + \dots + (2 \cdot (N-1) + 1)$$

quindi si utilizzeranno almeno 3 variabili:

N , i , S rispettivamente per il numero, il contatore e l' accumulatore.



Esercizio 2

Algoritmo - Linguaggio Informale

1. Inizio
2. Leggi un numero intero positivo dallo standard input.
3. Inizializza un contatore i a 0
4. Inizializza un accumulatore S a 0
5. Finché minore del numero letto
 1. Somma all'accumulatore il doppio del valore del contatore incrementato di 1
 2. Incrementa il contatore
 3. Torna al punto 5.
6. Stampa a terminale il valore dell'accumulatore
7. Fine

Esercizio 3

Si definisce **Triangolare** un numero costituito dalla somma dei primi N numeri interi positivi per un certo N .

Esempio:

per $Q = 10$

si ha $Q = 1 + 2 + 3 + 4$, da cui $N=4$.

o, utilizzando soltanto operazioni tra numeri interi.

In caso affermativo stampare a video il numero inserito e il massimo degli addendi che lo compongono.

Esercizio 3

Algoritmo - Linguaggio Informale

Idea: se $Q = 1 + 2 + 3 + \dots + i + \dots + N == 0$ per un certo N allora Q è Triangolare.

1. Leggi il numero positivo Q dallo standard input
2. Inizializza un contatore i a zero
3. Memorizza in una variabile S il valore della variabile in ingresso.
4. Finché maggiore di zero
 1. Incrementa di 1 il valore del contatore
 2. Sottrai a S il valore del contatore i
 3. Torna a 4
5. Se il valore residuo di S è zero allora
 1. Il numero è triangolare
 2. il valore del massimo degli addendi è uguale al contatore i
 3. la variabile Q contiene il valore della variabile in ingresso
6. Altrimenti il numero non è triangolare

Esercizio 4

Scrivere un programma che legge da stdin una sequenza (di lunghezza a priori illimitata) di numeri interi positivi, terminata da 0, e indica, alla fine della sequenza, qual è la lunghezza della massima sotto sequenza di numeri consecutivi in ordine crescente.

→ → →
13 3 8 4 5 1 17 0
Lung. max = 2

21 19 18 14 9 6 4 3 0
Lung. max = 1

→ →
2 1 3 6 8 5 1 12 18 17 0
Lung. max = 4

Esercizio 6

Scrivere un programma che legge un intero positivo n da stdin e verifica se può essere scomposto nella somma di due quadrati (verifica cioè se a, b, n sono tali che $a^2 + b^2 = n$). Se sì, stampare a video la scomposizione.

$$\mathbf{2 ==> 2 = 1 + 1 = 1^2 + 1^2}$$

$$\mathbf{28 ==> NON SCOMPONIBILE}$$

$$\mathbf{146 ==> 146 = 25 + 121 = 5^2 + 11^2}$$

a) Mostrare, quando ve ne è più di una, tutte le diverse scomposizioni dello stesso numero (ad esempio 50 ha due scomposizioni, $1+49$ e $25+25$, mentre 5525 è il primo numero ad avere ben sei diverse scomposizioni e 8125 è il primo ad averne esattamente cinque).

b) Verificare anche la scomponibilità in somma di tre quadrati.

