



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Informatica

Modulo di Programmazione

INFORMATICA
MODULO DI PROGRAMMAZIONE
C++ BASICS - SMALL APPLICATIONS

mauro.pelucchi@gmail.com

Mauro Pelucchi

2023/2024

Agenda

- Array
- Small Applications

Array

- Consecutive group of memory locations
 - Same name and type (int, char, etc.)
- To refer to an element
 - Specify array name and position number (index)
 - Format: `arrayname[position number]`
 - First element at position 0
- N-element array `c`
 - `c[0], c[1] ... c[n - 1]`
 - Nth element as position N-1

Array

- When declaring arrays, specify
 - Name
 - Type of array —> Any data type
- Number of elements
 - `type arrayName[arraySize];`
 - `int c[10]; // array of 10 integers`
 - `float d[3284]; // array of 3284 floats`
- Declaring multiple arrays of same type
 - Use comma separated list, like regular variables
 - `int b[100], x[27];`

Examples Using Arrays

- Initializing arrays
 - For loop
 - Set each element
 - Initializer list
- Specify each element when array declared
 - `int n[5] = { 1, 2, 3, 4, 5 };`
 - If not enough initializers, rightmost elements 0
 - If too many syntax error
- To set every element to same value
 - `int n[5] = { 0 };`
- If array size omitted, initializers determine size
 - `int n[] = { 1, 2, 3, 4, 5 };`
 - 5 initializers, therefore 5 element array

Esercizio 1

Si acquisisca da tastiera una sequenza (a priori illimitata) di caratteri terminata da **'\n'**.

Per poter acquisire ed elaborare anche eventuali caratteri di spaziatura, è necessario utilizzare il metodo **cin.get(...)**.

Si controlli se nella sequenza le parentesi tonde sono correttamente annidate e bilanciate, cioè se sono tutte ordinatamente richiuse.

Esercizio 1

Esempi:

Sequenze corrette:

ciao

()

(ciao)mondo

((ciao)(cia(o))(c(i)ao))(mond(o))

Sequenze non corrette:

cia)o

c(iao

ci(ao)mo(ndo

ci(ao)m(ond(o)!!

ciao(ci)ao)(mondo)

Suggerimento:

si tenga conto man mano di quante sono le parentesi già aperte, e si decida di conseguenza. Si arresti la scansione non appena si scopre che la sequenza è illegale, anche prima di arrivare alla fine (come succede, ad esempio, nel primo e nell'ultimo dei casi scorretti mostrati sopra; negli altri tre casi occorre invece arrivare alla fine della sequenza per poterla dichiarare scorretta).

Esercizio 2

Scrivere un programma che legge da stdin un numero intero positivo n e stampa a video le prime n righe del triangolo di Floyd (un triangolo rettangolo che contiene tutti i numeri naturali disposti come segue).

Ad esempio, per $n=10$:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55
```


Esercizio 2

La soluzione più immediata utilizza due cicli annidati. Si progetti, come esercizio di "elasticità espressiva", anche una soluzione che usa un solo ciclo.

Variante (probabilmente a prima vista non banale)

Si stampi il triangolo partendo dall'ultima riga (nell'es.: da 46):

46 47 48 49 50 51 52 53 54 55

37 38 39 40 41 42 43 44 45

29 30 31 32 33 34 35 36

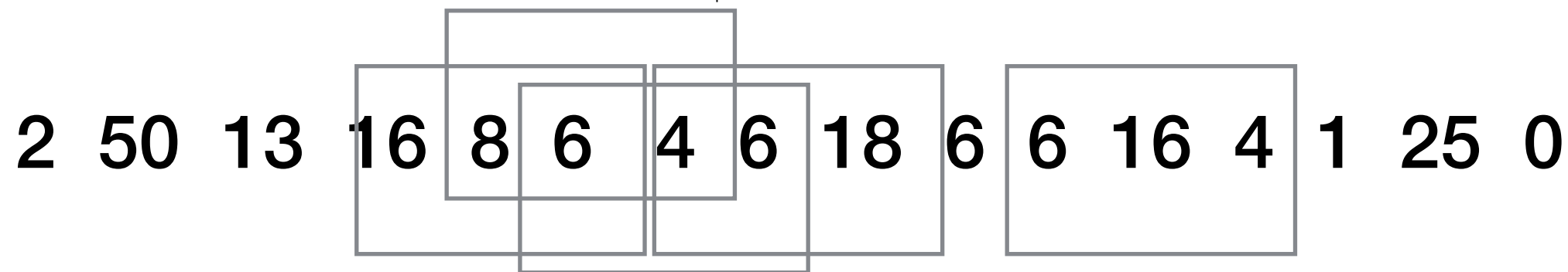
22 23 24 25 26 27 28

...



Esercizio 3

Si codifichi un programma C che legge dallo standard input una sequenza (di lunghezza arbitraria) di interi positivi terminata dal valore 0 e, al termine della sequenza, visualizza su standard output un messaggio che indica quante terne di numeri consecutivi diversi e pari sono contenute nella sequenza.



Le terne di numeri cercati sono 4.

Si noti che uno o due numeri potrebbero appartenere a più di una terna e che lo 0 è un numero pari che potrebbe far parte di una terna valida.

Esercizio 4

Scrivere un programma in linguaggio C che risolva il problema seguente.

Leggere un polinomio di grado n a coefficienti reali e valutarlo in un dato punto x .

Esercizio 5

Scrivere un programma che acquisisca una sequenza di 10 interi positivi e visualizzi l'istogramma corrispondente, come illustrato di seguito.

Variante: considerare anche valori numeri negativi o nulli.

Element	Value	Histogram
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****
5	9	*****
6	13	*****
7	5	*****
8	17	*****
9	1	*

Esercizio 5

```
#include <iostream>
#include <iomanip>
using namespace std;
int main () {
    ...
    int a = 13, n = 4;
    ...
    cout << setw(n) << a << setw(n) << a+1;
    ...
}
```

cout << setw(7) << a; riporta sul monitor il valore della variabile a utilizzando almeno 7 spazi.

In altre parole: cout << setw(7) << a; scriverà sul monitor _____13

Diversamente: cout << setw(5) << a; scriverebbe sul monitor ____13

Mentre: cout << a; scriverebbe sul monitor 13



Esercizio 6

Scrivere un programma che visualizzi (come illustrato in calce) i valori di un array di 10 interi qualsiasi (cioè anche negativi o nulli).

