* ***Bit***: misura minima memorizzata in un computer che contiene solo 2 valori: 0 o 1
* ***Byte***: 8 bit raggruppati assieme
* Il c++ è un ***linguaggio tipizzato*** ossia è necessario specificare la tipologia della variabile:
  + Int (long) – numeri interi
  + Float (double) – numeri con dei decimali
  + Char (String) – carattere
  + Bool – Valore booleano True o False
* ***Allocare una variabile***: assegnare ad essa una posizione nella RAM (formata da vari byte ognuno composto da 8 bit)
* Int8: variabile con un intero che può andare da 0 a 255
* Int16: variabile che occupa 2 byte
* Int32: variabile usata generalmente da tutti i linguaggi interpretati

Per inserire i numeri negativi si usano variabili signed o unsigned 🡪 quelle signed spostano la metà dei valori disponibili sotto lo 0

* ***Overflow***: quando si supera il valore massimo di un int e il contenuto diventa negativo.

Le assegnazioni di variabili primitive:

* Valore: ad una variabile viene assegnato un valore uguale a quello dell’altra. (n1 = a)
* Riferimento: quando una variabile è solo un altro modo per chiamarne un’altra (n1 &= a)

Void davanti ad una funzione si usa quando si sa che essa non restituirà nulla

* ***Prototipazione***: quando una funzione viene nominata all’inizio ma scritta alla fine.

Int somma (int add1, int add 2);

int main(){…}

Int somma (int add1, int add 2);

{“tutto ciò che fa la funzione”}

* ***Operatore ternario***: operatore che semplifica un if dentro una funzione

return (n>=0? n : -n);

***I puntatori***

* ***Puntatore***: ulteriore tipo di variabile definita come contenitore di un dato e quindi è una zona di memoria (0x…). I puntatori sono variabili che contengono l’indirizzo di memoria di una cella dei memoria.

Int p=0; Int \*p;

***Le strutture***

definire una struttura:

struct punto

{

int x;

int y;

};

struct punto p1;

p1.x = 10;

p1.y = 20;

***Gli array e le matrici (array bidimensionali)***

* ***Array/vettore***: è un aggregato di dati con all’interno dati uniformi in quantità definita durante la sua dichiarazione.
* ***Dimensione fisica***: dimensione di allocazione dell’array.
* ***Dimensione logica***: dimensione utilizzata discrezionalmente dell’array.
* ***Lista***: è come il vettore solo che in memoria gli elementi non hanno posizioni contigue ma sono celle collegate da puntatori. Se è bidirezionale ogni elemento ha un riferimento al precedente ed al successivo.
* ***Vector***: strutture dinamiche che collegano più vettori permettendo di ampliarli

Int v[] = {10, 21, 23, 11};

cout << \*v[0] 🡪 10 cout << \*(v+1) 🡪 21

O = complessità computazionale

complessità computazionale ricerca binaria O(log(n))

complessità computazionale ordinamento ingenuo O(n2)

complessità computazionale bubble sort caso peggiore O(n2)

time(NULL) 🡪 prende i secondi trascorsi dal 1974