

ASD Laboratorio 01

The A(SD)-Team

UniTN

2023-10-03

ESERCITATORI

- Quintino Francesco Lotito (`quintino.lotito@unitn.it`)
- Cristian Consonni (`cristian.consonni@unitn.it`)

TUTOR

- Carmen Casulli (`carmen.casulli@studenti.unitn.it`)
- Michele Ghilardi (`michele.ghilardi@studenti.unitn.it`)
- Luca Mosetti (`luca.mosetti-1@studenti.unitn.it`)

SITI INTERNET

- **Sito laboratorio (slides caricate in giornata):**
`http://judge.science.unitn.it/slides/`
- **Judge:** `http://judge.science.unitn.it`
- **Registrazione su Judge:**
`http://judge.science.unitn.it/registration`

CANALI DI COMUNICAZIONE

- **Canale Telegram del corso:** `https://t.me/unitn_asd23`
- **Canale Telegram per il tutorato:** `https://t.me/tutoratoASD23`

03/10	Introduzione
17/10	Ad-hoc
14/11	Grafi 1
28/11	Grafi 2
11/12	Presentazione Progetto 1
12/12	Lab Progetto 1
14/12	Lab Progetto 1

PROGETTO GRAFI

- Dal 11 al 18 dicembre (consegna ore 18:00);
- Iscrizione dei gruppi al progetto entro **venerdì 8 dicembre**:
https://noclick.dev/ASDprog_2022-2023 (dovete essere loggati con l'account UniTN)

PERCHÉ FARE UN LABORATORIO

WHO CAN EXPLAIN
frogSort ALGORITHM.



START WITH EMPTY LIST.
FOR EACH INTEGER, PUT
THAT NUMBER DEAD FLIES
IN ONE BOX. THEN PUT FROG
IN EACH BOX. WHEN FROG
LEAVE BOX, APPEND THAT
BOX'S FLY NUMBER
TO LIST.



VERY GOOD. NOW,
HOMEWORK IS PROGRAM
frogSort ON HOME FROGPUTER.



MORE FLY TAKE
LONGER TO EAT.
WHEN ALL FROGS
GONE FROM BOXES,
LIST ORDERED.



WHAT IS MAXIMUM
STEP NUMBER?

$\log_{ing}(\text{boxes})$.



LATER...

BAH! ME UNDERSTAND
BUT KEEP GETTING
OFF-BY-FROG ERROR!



Credits: SMBC comics

DA PSEUDOCODICE A CODICE

"You can't just copy-pase pseudocode into a program and expect it to work"



What's the point in declaring the data type of a variable anyway?

Credits: me.me

DA PSEUDOCODICE A CODICE

"You can't just copy-pase pseudocode into a program and expect it to work"



What's the point in declaring the data type of a variable anyway?

Credits: me.me

Per i laboratori useremo il C++:

- Fortemente tipizzato (+ auto)
- Compilato
- STL

OBIETTIVI DEL LABORATORIO

CAPACITÀ

Sapere la differenza fra pseudocodice e chiacchiere

Utilizzare i concetti imparati a lezione

Saper valutare l'efficienza di un algoritmo

ATTIVITÀ

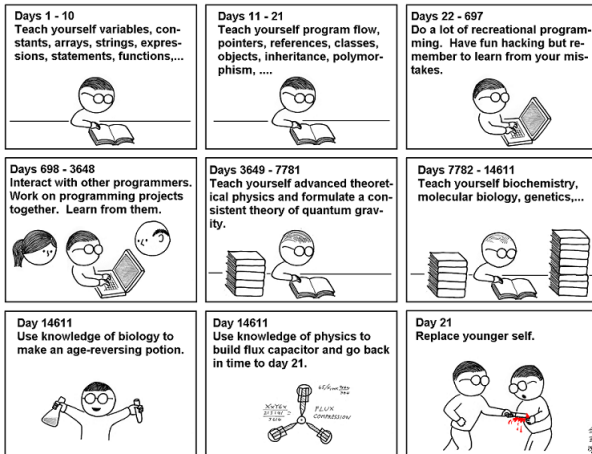
Passaggio da pseudocodice a codice

Risoluzione di problemi

Test automatizzato usando dati di differenti dimensioni

Useremo la Standard Template Library di C++ in modo da evitare la reimplementazione di strutture dati conosciute.

ARS LONGA, VITA BREVIS



As far as I know, this is the easiest way to "Teach Yourself C++ in 21 Days".

PROGAMMARE È UN'ARTE



CommitStrip.com

Credits: CommitStrip.com

SCRIVERE CODICE COMPRENSIBILE (I)

```
float _x = abs(x - deviceInfo->position.x) / scale;
int directionCode;
if (0 < _x && x != deviceInfo->position.x) {
    if (0 > x - deviceInfo->position.x) {
        directionCode = 0x04 /*left*/;
    } else if (0 < x - deviceInfo->position.x) {
        directionCode = 0x02 /*right*/;
    }
}
```

SCRIVERE CODICE COMPRENSIBILE (II)

```
static const int DIRECTIONCODE_RIGHT = 0x02;
static const int DIRECTIONCODE_LEFT = 0x04;
static const int DIRECTIONCODE_NONE = 0x00;

int oldX = deviceInfo->position.x;
int directionCode = (x > oldX)? DIRECTIONCODE_RIGHT
                    : (x < oldX)? DIRECTIONCODE_LEFT
                    : DIRECTIONCODE_NONE;
```



Credits: belief driven design

SCRIVERE BUONI COMMENTI (I)

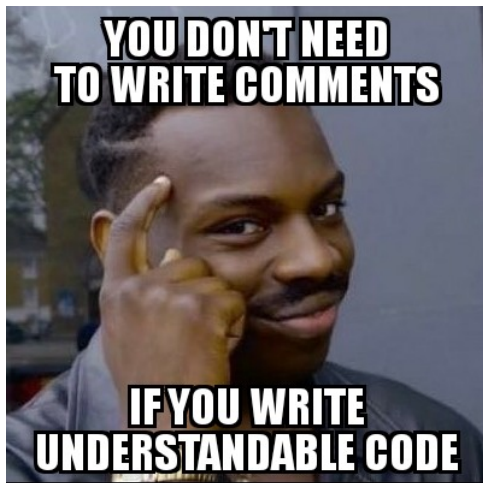
```
r = n/2;
while ( abs( r - (n/r) ) > t ) {
    r = 0.5 * ( r + (n/r) );
}
cout << "r = " << r << endl;
```

SCRIVERE BUONI COMMENTI (II)

```
// square root of n with Newton approximation
r = n/2;
while ( abs( r - (n/r) ) > t ) {
    r = 0.5 * ( r + (n/r) );
}
cout << "r = " << r << endl;
```

SCRIVERE BUONI COMMENTI (III)

```
double SquareRootApproximation(double n, double
    threshold) {
    double r = n/2;
    while ( abs( r - (n/r) ) > threshold ) {
        r = 0.5 * ( r + (n/r) );
    }
    return r;
}
cout << "r = " << SquareRootApproximation(n, threshold)
    << endl;
```

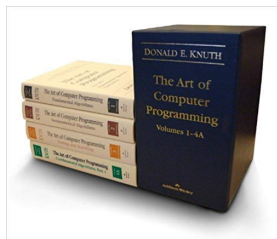



Credits: devRant

SCRIVETE CODICE PER GLI UMANI, NON PER LE MACCHINE

“Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.”

Donald Knuth



Lettere consigliate:

- Code Tells You How, Comments Tell You Why
- Coding Without Comments
- When Good Comments Go Bad
- Your comments are bad and you should feel bad

NON OBIETTIVI

- Ottimizzazioni a basso livello

SCRIVETE COSÌ

```
float f=...  
f*=pow(2,n);
```

NON COSÌ

```
float f=...  
if (*(int*)&f & 0x7FFFFFFF) {  
    *(int*)&f += n << 23;  
}
```

“We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil”

Donald Knuth

- Soluzioni dei problemi del laboratorio precedente (con consegna sorgenti)
- Descrizione di 3 o 4 problemi:
 - ▶ Traduzione da pseudocodice a codice
 - ▶ Problema semplice
 - ▶ Problema complicato
 - ▶ Vecchio progetto (non tutte le settimane)

⇒ di solito i problemi sono ordinati in modo tale che risolvendo i primi trovate delle idee per risolvere i successivi
- Lavoro individuale o gruppo per il resto del laboratorio (siamo qui per darvi una mano!)

CMS: CONTEST MANAGEMENT SYSTEM

Creato per l'edizione 2012 delle olimpiadi internazionali d'informatica

FUNZIONAMENTO

- Per ogni problema il sistema ha dei file di input ed una soluzione “ufficiale”
- Le vostre soluzioni devono leggere i dati di input da `input.txt` e scrivere su `output.txt`
- Il sistema riceve il sorgente e lo esegue per ogni file di input con un time limit e un memory limit per il singolo caso
- La soluzione riceve un punteggio da 0 a 100, in base a quante volte ha scritto la risposta corretta in tempo

⇒ Classifica su <https://judge.science.unitn.it/ranking>

ESEMPIO DI SOLUZIONE

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

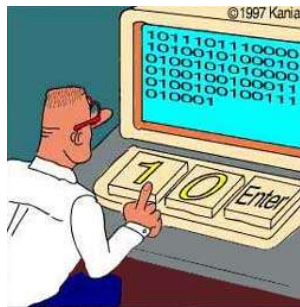
int main() {
    int N, M;
    ifstream in("input.txt");
    in >> N >> M;
    ofstream out("output.txt");
    out << N + M << "\n";
    return 0;
}
```

CMS: CONTEST MANAGEMENT SYSTEM

- **Accessibile da:** <http://judge.science.unitn.it>
- **Nome utente/password su:**
<http://judge.science.unitn.it/registration>
- **Sorgenti in C/C++**

SISTEMA DI SVILUPPO

...Quello che preferite! Per guide e consigli vedete il messaggio sul canale Telegram e/o chiedeteci.



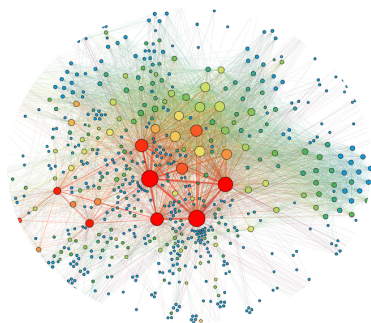
Real programmers code in binary.

- 1 progetto in questo semestre;
- 1 progetto nel prossimo semestre;
- Gruppi da 1, 2 o 3 persone;
- 8.5 giorni di tempo (~ 200 h);
- Sottoposizione usando CMS;
- Il progetto è superato se la soluzione fa almeno 30 punti su 100;
- Iscrizione su

[https:](https://noclick.dev/ASDprog_2022-2023)

[//noclick.dev/ASDprog_2022-2023](https://noclick.dev/ASDprog_2022-2023)

(dovete essere loggati con l'account UniTN).



- È necessario superare almeno un progetto(*) per accedere allo scritto
- I progetti completati durante il corso danno punti bonus allo scritto
- Il punteggio è assegnato in maniera competitiva
- Il progetto non è una barriera aggiuntiva

(*) per chi fa solo il primo modulo è necessario superare il primo

- È vietata collaborazione di alcun tipo fra i gruppi
- Potete chiedere agli assistenti in caso di difficoltà
- Abbiamo potenti mezzi e li usiamo
- Copiando guadagnate al massimo 1/2 punti allo scritto
- Se vi becchiamo...



NOTE DI COMPILAZIONE

- Sul server viene usato -DEVAL
- C o C++, ma usate C++ per le STL
- Fate riferimento allo standard C++11

I nostri esempi saranno C++11 (compilare con `-std=c++11`)

STANDARD TEMPLATE LIBRARY

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

Documentazione online (anche su judge)

<http://www.cplusplus.com/reference/>

LETTURA DA FILE

- Usate `ifstream`
- Usate `cin`, riconosce il tipo delle variabili passate ed ignora spazi ed invii

LETTURA INPUT

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

int main(){
    ifstream in("input.txt");
    int N;
    in >> N;
    for(int i=0; i < N; i++){
        int a;
        in >> a;
    }
    ...
}
```

SCRITTURA SU FILE

- Usate `ofstream`
- Usate `cout`, riconosce il tipo delle variabili passate

SCRITTURA OUTPUT

```
...
ofstream out("output.txt");
out << N << endl;
for(int el:vec) {
    out << el << endl;
}
return 0;
}
```

Equivalente all'arraylist di java.

```
#include <vector>
//Crea vector di interi
vector<int> intvec;
//Crea vector di 7 float inizializzati a 0.5
vector<float> floatvec(7,0.5);
//Accedi agli elementi
floatvec[2] = floatvec[5] + 0.1;
//Aggiungi un elemento in fondo al vector
intvec.push_back(231);
//Cicla sugli elementi:
for(int i=0; i < intvec.size(); i++)
    intvec[i] = 12;
//Ridimensiona vector
intvec.resize(100);
```

Coppia di elementi.

```
#include <utility>
//pair di intero e float
pair<int, float> coppial;
//assegnazione elementi
coppial.first = 2;
coppial.second = 3.4;
coppial = make_pair(15, 0.4);
//coppia di coppie
pair< pair<int, int>, pair<int, int> > c;
```

```
#include <algorithm>
//ordinare un array di N elementi
sort(arr, arr + N);
//ordinare un vector
sort(vec.begin(), vec.end());
```


CODING: SORTING STRUCTS

```
#include <algorithm>
#include <vector>
using namespace std;

struct stud {
    int id;
    int voto;
};

bool operator < (const stud a, const stud b){
    return a.voto < b.voto;
}

int main() {
    vector<stud> arr(2);
    arr[0].id=1; arr[0].voto=30;
    arr[1].id=2; arr[1].voto=20;
    sort(arr.begin(), arr.end());
}
```

```
#include <queue>
//Dichiarare coda di interi
queue<int> q;
//Aggiungere un elemento alla coda
q.push(23);
//Leggere l'elemento in testa alla coda
int el = q.front();
//Eliminare l'elemento in testa alla coda
q.pop();
//Controllare se la coda e vuota
if(q.empty())
    ...
```

```
#include <stack>
//Dichiarare pila di interi
stack<int> s;
//Aggiungere un elemento in cima alla pila
s.push(23);
//Leggere l'elemento in cima alla pila
int el = s.top();
//Eliminare l'elemento in cima alla pila
s.pop();
//Controllare se la pila e vuota
if(s.empty())
    ...
```

NOTE SU C++11 (I)

- For-each
- auto

```
vector<int> arr = ...;
for(int el:arr){
    cout << el << endl;
}
for(int& el:arr){
    el++;
}
auto d = 23;
for(auto& el:arr){
    el += d;
}
return arr;
```

- Move operator

```
template<class T> void swap(T& a, T& b) {  
    T tmp { std::move(a) };  
    a = std::move(b);  
    b = std::move(tmp);  
}
```

SOMMA DI DUE NUMERI

Dati due interi, sommateli.

INPUT.TXT

Due interi N, M separati da spazio

OUTPUT.TXT

Un intero, uguale alla somma di N e M.

Esempio:

```
input.txt
```

```
2 3
```

```
output.txt
```

```
5
```

SOTTOSEQUENZA DI SOMMA MASSIMA

Data una sequenza di interi, trovare la sottosequenza di somma massima.

INPUT.TXT

N+1 righe: il numero di elementi N sulla prima riga e gli N elementi nelle N righe seguenti.

Esempio:

```
input.txt
```

```
5
3
-2
4
1
5
```

```
output.txt
```

```
11
```

SOTTOMATRICE DI SOMMA MASSIMA

Data una matrice di interi, trovare la sottomatrice di somma massima.

INPUT.TXT

R+1 righe: R e C (numero di righe e di colonne) sulla prima riga, C interi su ognuna delle seguenti R righe.

Esempio:

```
input.txt
```

```
3 4
2 -9 2 3
1 4 5 1
-2 3 4 1
```

```
output.txt
```

```
18
```


- 1 Se non avete già un account:
`http://judge.science.unitn.it/registration`
- 2 Implementate una soluzione per il problema della somma e testatela su
`http://judge.science.unitn.it`
- 3 Risolvete gli altri problemi
- 4 Non usate judge come compilatore!

NOTE

- I file C++ devono avere l'estensione `.cpp`