Programmazione a moduli

Modulo sorgente / modulo oggetto / modulo eseguibile

Scrittura: usare un editor e salvare
modulo1.cpp

Compilazione:

Compilare un modulo.cpp e generare modulo.o

```
g++ -c modulo1.cpp -o modulo1.o
```

Linking:

Collegare il singolo modulo con altri moduli oggetto che contengono codice utile a generare il modulo eseguibile.

```
g++ modulo1.o modulo2.o -o modulo.exe
```

Esecuzione:

dal prompt dei comandi

```
./modulo.exe
```

```
modulo.exe
```

Modulo sorgente

```
// modulo1.cpp

int somma(int a, int b){
int c = a+b;

return c;
}
```

Compilazione a moduli

```
//modulo2.cpp
#include <iostream>
using namespace std;

int somma(int, int);

int main() {
int x, y, z;
cin >> x;
cin >> y;
z = somma(x, y);
cout << z;
return 0;
}
```

```
g++ -c modulo1.cpp -o modulo1.o
```

```
g++ -c modulo2.cpp -o modulo2.o
```

```
g++ modulo1.o modulo2.o -o modulo.exe
```

Libreria matematica

- `cmath.h`

Dichiarazione funzioni

```
double pow(double base, double exp);
```

.....

- `cmath`

La libreria contiene il codice oggetto di tutte le funzioni. Codice già compilato. Il sorgente `.cpp` non è disponibile

```
//modulo.cpp
#include <iostream>
#include <cmath>
using namespace std;
```

```
int main(){
double x, y, z;
cin >> x;
cin >> y;
z = pow(x,y);
cout << z;
return 0;
}
```

```
g++ -c modulo.cpp -o modulo.o
```

Nella fase di compilazione (opzione -c) la direttiva #include viene sostituita con il contenuto del file. Viene generato modulo oggetto.

```
g++ modulo.o -l math -o modulo.exe
```

Nella fase di linking (manca opzione -c) -l math indica al linker che devono essere collegate funzioni della libreria matematica. Alcune librerie non e' necessario specificarle (per esempio math) Posso scrivere:

```
g++ modulo.o -o modulo.exe
```