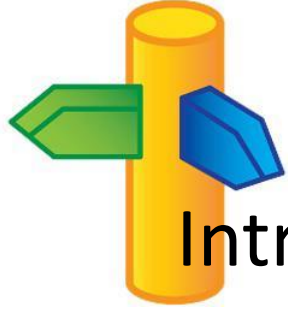




# UNIVERSITÀ DELLA CALABRIA



Introduzione alle funzionalità del pacchetto

## IBM ILOG CPLEX Optimization Studio

A cura di:

**Cristian Belfiore**, Dottore Magistrale in Ingegneria Gestionale

**de-Health Lab - Laboratory of Decision Engineering for Health Care Services**

**Ponte Pietro Bucci 41C, 8° piano - 87036 Rende (Cosenza)**

mail: [cristian.belfiore@unical.it](mailto:cristian.belfiore@unical.it)



UNIVERSITÀ DELLA CALABRIA   
DIPARTIMENTO DI  
INGEGNERIA MECCANICA,  
ENERGETICA E GESTIONALE  
DIMEG

## IBM ILOG CPLEX Optimization Studio

### PROBLEMA 2

Una ASL ha attivato, in alcune zone rurali del proprio territorio di competenza, 2 nuove postazioni per il prelievo di campioni di sangue. Per essere esaminati, i campioni vanno inviati presso i 3 laboratori di analisi dei presidi ospedalieri di competenza della stessa ASL e situati in punti diversi del territorio.

In relazione alle caratteristiche tecniche ed al personale disponibile, le postazioni sono in grado di effettuare, rispettivamente, al più 250 e 400 prelievi al mese. Inoltre, i laboratori di analisi, sulla base dei loro flussi di lavoro, sono in grado di esaminare, rispettivamente, 120, 170 e 130 campioni.

Per il trasporto dei campioni dalle postazioni di prelievo ai laboratori, l'ASL deve sostenere dei costi, i cui valori sono riportati nella seguente tabella.

	Laboratori		
Postazioni	2	4	1
Prelievo	3	6	5

L'obiettivo principale che si pone l'ASL è quello di effettuare il trasporto dei campioni minimizzando i costi complessivi.

```

1 /*****
2  * OPL 22.1.0.0 Model
3  * Author: Win10
4  * Creation Date: 1 mag 2023 at 14:03:56
5  *****/
6 {string} Postazioni = ...;
7 {string} LaboratoriAnalisi = ...;
8
9 int CapacitaPostazioni [Postazioni] = ...;
10 int CapacitaLaboratori [LaboratoriAnalisi] = ...;
11
12 int CostiTrasporto [Postazioni][LaboratoriAnalisi] = ...;
13
14 dvar int+ CampioniTrasportati [Postazioni][LaboratoriAnalisi];
15
16
17
18 minimize sum (i in Postazioni, j in LaboratoriAnalisi) CostiTrasporto[i][j]*CampioniTrasportati[i][j];
19
20
21
22 subject to{
23
24 forall (i in Postazioni)
25     sum (j in LaboratoriAnalisi) CampioniTrasportati[i][j] <= CapacitaPostazioni [i];
26
27 forall (j in LaboratoriAnalisi)
28     sum (i in Postazioni) CampioniTrasportati[i][j] == CapacitaLaboratori [j];
29 }
```

```

1 /*****
2  * OPL 22.1.0.0 Data
3  * Author: Win10
4  * Creation Date: 1 mag 2023 at 14:04:44
5  *****/
6 Postazioni = {"Postazione1", "Postazione2"};
7 LaboratoriAnalisi = {"Laboratorio1", "Laboratorio2", "Laboratorio3"};
8
9 CapacitaPostazioni =[250, 400];
10
11 CapacitaLaboratori =[120, 170, 130];
12
13 CostiTrasporto= [[2, 4, 1]
14                  [3, 6, 5]];
```

## IBM ILOG CPLEX Optimization Studio

### PROBLEMA 5

Una grande Azienda Farmaceutica è specializzata nella produzione di un nuovo farmaco per pazienti affetti da scompenso cardiaco cronico.

Dal punto di vista della distribuzione, il farmaco viene fornito ai centri di smistamento in modo periodico, alla fine di ogni mese. In particolare, le richieste  $d_t$  (esprese in unità di prodotto) vengono organizzate in modo da coprire un orizzonte temporale di 5 mesi, come mostrato nella seguente tabella:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
$d_t$	270	290	250	240	310

Per la produzione del farmaco si utilizza una linea di produzione a ciclo continuo caratterizzata da costi unitari di produzione mensile  $c_t$  (espressi in migliaia di Euro) e da una capacità di produzione mensile  $b_t$  (esprese in unità di prodotto), così come descritto nella seguente tabella:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
$c_t$	25	28	32	21	24
$b_t$	250	220	280	270	260

I farmaci eventualmente non consegnati saranno immagazzinati. Ciò comporta un costo aggiuntivo di inventario, il cui valore unitario  $h_t$  (espresso in migliaia di Euro) varia mensilmente nel seguente modo:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
$h_t$	10	12	8	10	11

In magazzino sono disponibili inizialmente 100 unità di prodotto. L'obiettivo dell'azienda è pianificare la produzione del farmaco in modo tale da minimizzare i costi di produzione e di inventario.

Formulare un modello di ottimizzazione che rappresenti e risolva adeguatamente il problema.

```

*Problema_5.mod  Problema_5.dat
4  * Creation Date: 1 mag 2023 at 15:05:53
5  *****/
6  int T=...;
7  range OrizzonteTemporale=1..T;
8
9  int domandaFarmaco[OrizzonteTemporale] = ...;
10
11 int costiProduzione[OrizzonteTemporale] = ...;
12
13 int capacitaProduzione[OrizzonteTemporale] = ...;
14
15 int costoStoccaggio [OrizzonteTemporale] =...;
16
17 int disponibilitaIniziale = ...;
18
19 dvar int+ quantitaProdotta[OrizzonteTemporale];
20 dvar int+ quantitaStoccata[OrizzonteTemporale];
21
22
23 minimize sum (t in OrizzonteTemporale)
24     ((quantitaProdotta[t]*costiProduzione[t]) + (quantitaStoccata[t]*costoStoccaggio[t]));
25
26
27 subject to {
28
29 vincoloTempo1: quantitaStoccata[1] == disponibilitaIniziale + quantitaProdotta[1] - domandaFarmaco[1];
30
31 forall (t in OrizzonteTemporale : t>1)
32     quantitaStoccata[t] == quantitaStoccata[t-1] + quantitaProdotta[t] - domandaFarmaco[t];
33
34 forall (t in OrizzonteTemporale)
35     quantitaProdotta[t] <= capacitaProduzione[t];
36 }
37
38 //COMMENTO
39 /*COMMENTO*/
  
```



### ***Problema 5 – uso di Excel***

```

1 /*****
2 * OPL 22.1.0.0 Data
3 * Author: Win10
4 * Creation Date: 1 mag 2023 at 16:56:01
5 *****/
6
7 SheetConnection Sheet1 ("Problema_5.xlsx");
8
9 T from SheetRead(Sheet1, "Dati!I6");
10
11 domandaFarmaco from SheetRead(Sheet1, "Dati!B4:F4");
12
13 costiProduzione from SheetRead(Sheet1, "Dati!B6:F6");
14
15 capacitaProduzione from SheetRead(Sheet1, "Dati!B8:F8");
16
17 costoStoccaggio from SheetRead(Sheet1, "Dati!B10:F10");
18
19 disponibilitaIniziale from SheetRead(Sheet1, "Dati!I4");
20
21
22 costo to SheetWrite(Sheet1, "Risultati!B9");
23
24 quantitaProdotta to SheetWrite(Sheet1, "Risultati!B4:G4");
25
26 quantitaStoccata to SheetWrite(Sheet1, "Risultati!B6:G6");

```

[illegible][illegible]



## IBM ILOG CPLEX Optimization Studio

```
Problema_6.mod  Problema_6.dat X
1 /*****
2 * OPL 22.1.0.0 Data
3 * Author: Win10
4 * Creation Date: 1 mag 2023 at 16:08:31
5 *****/
6 Farmaci = {"A", "B", "C"};
7
8 Stabilimenti = {"S1", "S2"};
9
10 costoLavorazione = [50, 75];
11
12 produttivita = [[3, 2, 3]
13                [4, 4, 5]];
14
15 domandaGiornaliera = [90, 120, 100];
16
17 produzioneSostanzeInquinanti = [20, 10, 10];
18
19 UBGiornaliero_SostanzeInquinanti = 6000;
20
21 disponibilitaGiornalieraImpianti = 24;

Problema_6.mod X  Problema_6.dat
1 /*****
2 * OPL 22.1.0.0 Model
3 * Author: Win10
4 * Creation Date: 1 mag 2023 at 16:08:31
5 *****/
6 {string} Farmaci = ...;
7 {string} Stabilimenti = ...;
8
9 int costoLavorazione [Stabilimenti] = ...;
10
11 int produttivita[Stabilimenti][Farmaci] = ...;
12
13 int domandaGiornaliera [Farmaci] = ...;
14
15 int produzioneSostanzeInquinanti [Farmaci] = ...;
16
17 int UBGiornaliero_SostanzeInquinanti = ...;
18
19 int disponibilitaGiornalieraImpianti = ...;
20
21 dvar float+ oreLavorazione[Stabilimenti];
22
23 minimize sum (s in Stabilimenti) costoLavorazione[s]*oreLavorazione [s];
24
25 subject to {
26
27 forall (f in Farmaci)
28     sum (s in Stabilimenti) oreLavorazione[s] * produttivita [s][f] >= domandaGiornaliera[f];
29
30 sum (s in Stabilimenti, f in Farmaci) (oreLavorazione[s] * produttivita [s][f] * produzioneSostanzeInquinanti [f]) <= UBGiornaliero_SostanzeInquinanti;
31
32 forall (s in Stabilimenti)
33     oreLavorazione[s] <= disponibilitaGiornalieraImpianti;
34 }
```

### PROBLEMA 6

Un'industria farmaceutica produce tre tipi di farmaci, A, B e C, utilizzando due stabilimenti di produzione. Nel primo impianto, in un'ora di lavorazione (il cui costo è di 50 Euro), si producono 3 Kg del farmaco A, 2 Kg del farmaco B e 3 Kg del farmaco C. Nel secondo impianto, un'ora di lavorazione ha un costo pari a 75 Euro e per ogni ora si producono 4 Kg di farmaco A, 4 Kg di farmaco B e 5 Kg di farmaco C.

Dei tre farmaci è richiesta una produzione giornaliera pari ad almeno 90, 120 e 100 Kg, rispettivamente. La produzione dei tre farmaci comporta la generazione di sostanze inquinanti; in particolare per ogni Kg di farmaco A si producono 20 grammi di sostanza inquinante, mentre per ogni Kg di farmaco B o C la quantità di sostanza inquinante prodotta è pari a 10 grammi. Una legge nazionale sul controllo della qualità dell'ambiente impone che non è possibile produrre giornalmente più di 5 kg di sostanze inquinanti.

Tenendo conto del fatto che gli impianti possono essere utilizzati a ciclo continuo, formulare un modello di Programmazione Lineare relativo alla pianificazione ottimale della produzione giornaliera dei tre farmaci, con l'obiettivo di minimizzare i costi complessivi di lavorazione.



## IBM ILOG CPLEX Optimization Studio

### Strutturare i dati: le Tuple

```
1 /*****  
2  * OPL 22.1.0.0 Model  
3  * Author: Win10  
4  * Creation Date: 31 gen 2023 at 11:15:32  
5  *****/  
6 {string} Rottami = ...;  
7  
8 tuple CaratteristicheRottame {  
9     float Ferro;  
10     float Nichel;  
11     float Cromo;  
12     float Impurita;  
13  
14     float PesoDisponibile;  
15     float Costo;  
16 }  
17  
18 CaratteristicheRottame Rottame [Rottami] = ...;
```

La **tuple** è un costrutto del linguaggio di grande utilità: essa permette di **definire un nuovo tipo di dato** avente la struttura definita dal programmatore

Definisce un vettore di nome **Rottame**, avente **|Rottami|** elementi, ognuno dei quali è di tipo **CaratteristicheRottame**, ovvero è una **tuple** avente le caratteristiche definite a monte, nella dichiarazione della stessa

```
19  
20 dvar float+ Quantita [Rottami];  
21  
22 minimize sum (r in Rottami) Quantita[r] * Rottame[r].Costo;  
23  
24 subject to {  
25  
26     sum (r in Rottami) Quantita[r] == 100;  
27  
28     sum (r in Rottami) Quantita[r] * Rottame[r].Nichel == 18;  
29  
30     sum (r in Rottami) Quantita[r] * Rottame[r].Cromo == 10;  
31  
32     sum (r in Rottami) Quantita[r] * Rottame[r].Ferro >= 65;  
33  
34     sum (r in Rottami) Quantita[r] * Rottame[r].Impurita <= 1;  
35  
36  
37 forall (r in Rottami)  
38     Quantita[r] <= Rottame[r].PesoDisponibile;  
39  
40 forall (r in Rottami)  
41     Quantita[r] >= 0;  
42  
43 }  
44  
45 1 /*****  
46 2  * OPL 22.1.0.0 Data  
47 3  * Author: Win10  
48 4  * Creation Date: 31 gen 2023 at 11:15:39  
49 5  *****/  
50 6  
51 7 Rottami = {"Rottame1", "Rottame2", "Rottame3", "Rottame4", "Rottame5", "Rottame6"};  
52 8  
53 9 Rottame =  
54 10 #[  
55 11 Rottame1 : < 0.93, 0.05, 0.00, 0.02, 30, 50 >,  
56 12 Rottame2 : < 0.76, 0.13, 0.11, 0.00, 90, 100 >,  
57 13 Rottame3 : < 0.74, 0.11, 0.12, 0.03, 50, 80 >,  
58 14 Rottame4 : < 0.65, 0.16, 0.14, 0.05, 70, 85 >,  
59 15 Rottame5 : < 0.72, 0.06, 0.20, 0.02, 60, 92 >,  
60 16 Rottame6 : < 0.68, 0.23, 0.08, 0.01, 50, 115 >  
61 17 ]#;
```

L'accesso alle proprietà di una **tuple** avviene attraverso l'istruzione **nomeTuple.nomeProprietà**

Estratto che mostra la sintassi da usare per **assegnare i valori** alle caratteristiche interne di una **tuple** nel file .dat



## IBM ILOG CPLEX Optimization Studio

\*Ansalmecc\_Model.mod    Ansalmecc.Data.dat    Dieta\_Tuple\_Model.mod ×

```
5 *****/
6 {string} Macronutrienti = ...;
7 {string} Pietanze = ...;
8
9 float LimitiMacro [Macronutrienti] = ...;
10 float QuantitaGiornaliera = ...;
11 float MaxGrassoGiornaliero = ...;
12
13 tuple CaratteristichePietanza {
14   float Proteine;
15   float Carboidrati;
16   float Grassi;
17   float Costo;
18 }
19
20 CaratteristichePietanza DatiPietanza [Pietanze] = ...;
21
22 dvar float+ Quantita [Pietanze];
23
24 minimize sum (p in Pietanze) Quantita [p] * DatiPietanza[p].Costo;
25
26 subject to {
27
28   sum (p in Pietanze) Quantita[p] == QuantitaGiornaliera;
29
30   sum (p in Pietanze) DatiPietanza[p].Proteine * Quantita[p] >= LimitiMacro["Proteine"];
31
32   sum (p in Pietanze) DatiPietanza[p].Carboidrati * Quantita[p] >= LimitiMacro["Carboidrati"];
33
34   sum (p in Pietanze) DatiPietanza[p].Grassi * Quantita[p] >= LimitiMacro["Grassi"];
35
36   sum (p in Pietanze) DatiPietanza[p].Grassi * Quantita[p] <= MaxGrassoGiornaliero;
37
38   (Quantita["C1"] + Quantita["C2"]) >= 2 * Quantita["C3"];
39
40 }
41
42 execute Scripting {
43   writeln(" ");
44
45   for (var p in Pietanze)
46     writeln ("Nella dieta sono previsti " + Quantita[p] + " g di alimento " + p + " giornalmente")
47
48 }
```

	Proteine	Carboidrati	Grassi	Costo
P1	0.3	0.4	0.2	0.8
P2	0.2	0.4	0.0	0.4
P3	0.3	0.0	0.2	1.5
LB	100	150	100	
UB			200	
Qgiornaliera	600			

```
*Ansalmecc_Model.mod    Ansalmecc.Data.dat    Dieta_Tuple_Model.mod    Dieta_Tuple_Data.dat ×
1 /*****
2 * OPL 22.1.0.0 Data
3 * Author: Cristian
4 * Creation Date: 30 gen 2023 at 12:51:26
5 *****/
6 Macronutrienti = {"Proteine", "Carboidrati", "Grassi"};
7 Pietanze = {"C1", "C2", "C3"};
8
9 LimitiMacro = #["Proteine":100, "Carboidrati":150, "Grassi":100]#;
10
11 QuantitaGiornaliera = 600;
12 MaxGrassoGiornaliero = 200;
13
14 DatiPietanza =
15 #[
16 C1 : < 0.3, 0.4, 0.2, 0.8 >,
17 C2 : < 0.2, 0.4, 0.0, 0.4 >,
18 C3 : < 0.3, 0.0, 0.2, 1.5 >
19 ]#;
20
```



## IBM ILOG CPLEX Optimization Studio

ESEMPIO

L'acciaio è uno dei prodotti più facilmente riciclabili (e riciclati) al mondo. In effetti, è sufficiente fondere qualsiasi rottame di ferro per incenerire tutti gli eventuali residui plastici o di vernice contenuti nel rottame, restando così con solo metallo liquido. Il problema nasce in quanto è difficile separare i diversi metalli presenti nel rottame, per cui, insieme al ferro, si ritrovano nel metallo liquido anche rame, nichel, cromo e altri metalli. In diverse produzioni alcuni metalli sono desiderati e altri no. Per esempio, nella produzione dell'acciaio 18/10 (utilizzato nella produzione di pentole), si vuole avere il 18% di cromo e il 10% di nichel nel prodotto finito, per cui l'eventuale presenza di questi metalli nei rottami di ferro è altamente desiderabile, in quanto cromo e nichel sono molto più costosi sia dei rottami, sia dello stesso acciaio 18/10. Al contrario, il rame è un'impurità che rovina le caratteristiche estetiche dell'acciaio 18/10.

Ansalmecc ha di recente analizzato le caratteristiche di sei lotti di rottami di ferro, riportate in Tabella 2.3. Nella stessa tabella sono riportati anche il peso complessivo di ciascun lotto e il costo unitario di acquisto. L'obiettivo per l'azienda è produrre, al costo minimo, almeno 100 quintali di acciaio 18/10 con una presenza del 18% di cromo, 10% di nichel, almeno il 65% di ferro e al più un 1% di impurità.

Componente	Ingrediente					
	Rottame 1	Rottame 2	Rottame 3	Rottame 4	Rottame 5	Rottame 6
	[%]	[%]	[%]	[%]	[%]	[%]
Ferro	93	76	74	65	72	68
Nichel	5	13	11	16	6	23
Cromo	0	11	12	14	20	8
Impurità	2	0	3	5	2	1
Peso [q]	30	90	50	70	60	50
Costo [€]	50	100	80	85	92	115

Tabella 2.3. Dati relativi al problema di miscelazione della Ansalmecc.

\*Ansalmecc\_Model.mod × Ansalmecc\_Data.d

```

1 /*****
2 * OPL 22.1.0.0 Model
3 * Author: Win10
4 * Creation Date: 31 gen 2023 at 11:15:32
5 *****/
6 {string} Rottami = ...;
7
8 tuple CaratteristicheRottame {
9   float Ferro;
10  float Nichel;
11  float Cromo;
12  float Impurita;
13
14  float PesoDisponibile;
15  float Costo;
16 }
17
18 CaratteristicheRottame Rottame [Rottami] = ...; // Crea vettore di Tuple CaratteristicheRottame di dimensione pari a cardinalità di [Rottami]
19
20 dvar float+ Quantita [Rottami];
21
22 minimize sum (r in Rottami) Quantita[r] * Rottame[r].Costo;
23
24 subject to {
25
26   sum (r in Rottami) Quantita[r] == 100;
27
28   sum (r in Rottami) Quantita[r] * Rottame[r].Nichel == 18;
29
30   sum (r in Rottami) Quantita[r] * Rottame[r].Cromo == 10;
31
32   sum (r in Rottami) Quantita[r] * Rottame[r].Ferro >= 65;
33
34   sum (r in Rottami) Quantita[r] * Rottame[r].Impurita <= 1;
35
36
37 forall (r in Rottami)
38   Quantita[r] <= Rottame[r].PesoDisponibile;
39
40 forall (r in Rottami)
41   Quantita[r] >= 0;
42
43 }
44
45 execute {
46
47   for (var r in Rottami)
48     writeln ("La quantità di rottame " + r + " è pari a " + Quantita[r]);
49 }
50
51 // La soluzione fornita, per errore, è calcolata imponendo il 10% di Cromo ed il 10% di Nichel.
52

```

```

2 * OPL 22.1.0.0 Data
3 * Author: Win10
4 * Creation Date: 31 gen 2023 at 11:15:39
5 *****/
6
7 Rottami = {"Rottame1", "Rottame2", "Rottame3", "Rottame4", "Rottame5", "Rottame6"};
8
9 Rottame =
10 #[
11 Rottame1 : < 0.93, 0.05, 0.00, 0.02, 30, 50 >,
12 Rottame2 : < 0.76, 0.13, 0.11, 0.00, 90, 100 >,
13 Rottame3 : < 0.74, 0.11, 0.12, 0.03, 50, 80 >,
14 Rottame4 : < 0.65, 0.16, 0.14, 0.05, 70, 85 >,
15 Rottame5 : < 0.72, 0.06, 0.20, 0.02, 60, 92 >,
16 Rottame6 : < 0.68, 0.23, 0.08, 0.01, 50, 115 >
17 ]#;
18
19
20

```





## IBM ILOG CPLEX Optimization Studio

### Rappresentare la soluzione: lo Scripting log

```
40 execute Visualizza {
41   for (var s in SitiProduttivi)
42     for (var m in Magazzini)
43       if (QuantitaSM[s][m] != 0)
44         writeln ("La quantità da trasportare dal sito produttivo di " + s + " al magazzino di " + m + " è di " + QuantitaSM[s][m] + " pezzi." )
45
46   writeln (" ");
47
48   for (var m in Magazzini)
49     for (var p in PuntiVendita)
50       if (QuantitaMP[m][p] != 0)
51         writeln ("La quantità da trasportare dal magazzino di " + m + " al punto vendita di " + p + " è di " + QuantitaMP[m][p] + " pezzi." )
52 }
```

// solution (optimal) with objective 31851000

La quantità da trasportare dal sito produttivo di Rotterdam al magazzino di Bristol è di 150000 pezzi.

La quantità da trasportare dal sito produttivo di Rotterdam al magazzino di Middlesbrough è di 140000 pezzi.

La quantità da trasportare dal magazzino di Bristol al punto vendita di Londra è di 90000 pezzi.

La quantità da trasportare dal magazzino di Bristol al punto vendita di Birmingham è di 60000 pezzi.

La quantità da trasportare dal magazzino di Middlesbrough al punto vendita di Birmingham è di 20000 pezzi.

La quantità da trasportare dal magazzino di Middlesbrough al punto vendita di Leeds è di 50000 pezzi.

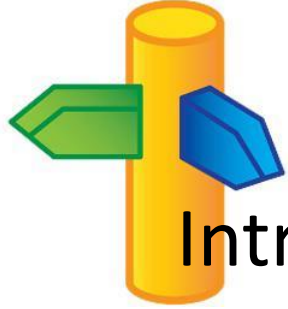
La quantità da trasportare dal magazzino di Middlesbrough al punto vendita di Edimburgo è di 70000 pezzi.

All'interno del blocco **execute**, tra l'altro, è possibile inserire del codice che permette di rappresentare la soluzione nella formattazione voluta dal programmatore.

Nelle figure il codice ed il risultato corrispondente.



# UNIVERSITÀ DELLA CALABRIA



Introduzione alle funzionalità del pacchetto

## IBM ILOG CPLEX Optimization Studio

A cura di:

**Cristian Belfiore**, Dottore Magistrale in Ingegneria Gestionale

**de-Health Lab - Laboratory of Decision Engineering for Health Care Services**

**Ponte Pietro Bucci 41C, 8° piano - 87036 Rende (Cosenza)**

mail: [cristian.belfiore@unical.it](mailto:cristian.belfiore@unical.it)



UNIVERSITÀ DELLA CALABRIA   
DIPARTIMENTO DI  
INGEGNERIA MECCANICA,  
ENERGETICA E GESTIONALE  
DIMEG