

```

!LISTATO DEL MODELLO
model Modellofinale_DUCA_BOGGIO_conRiserva
uses "mmxprs", "mmive";
!definizione dei file dei dati di ingresso
parameters
    fileCorsi = "semestrel_corsi_triennale-magistrale.txt"
    fileLaib = "semestrel_laib_duca-boggio.txt"
    !Aggiungo anche il file dove definisco il modulo orario, che qui chiamo
    "slot"
    fileSlot= "codiceslot.txt"
    !numSlot = 34
    numCorsida3 = 66
    numCorsida2 = 168
    !output file
    outrisultato = "risultato modello finale_DUCA-BOGGIO.xls"
    outCorsiAssegnati = "corsi assegnati_DUCA-BOGGIO.xls"
    outCorsiNonAssegnati = "corsi non assegnati_DUCA-BOGGIO.txt"
    outLaibrimasti = "laib rimasti liberi_DUCA-BOGGIO.xls"
!inserisco il file dei corsi che non sono riuscita ad assegnare e che devo avere
un orario
!di riserva
    fileCorsiRiserva = "corsi da riassegnare.txt"
    numCorsida3Riserva = 0
    numCorsida2Riserva = 2
!output file dei Corsi riassegnati con orario di riserva
    outfileRiserva = "risultato modello finale_corsiRiserva.txt"
end-parameters
!definizione delle variabili e dei dati contenuti nei file

declarations
!definisco i dati del file che contiene i Corsi
    numCorsi: integer
    nomeCorso: array(range) of string
    pesoCorso: array(range) of integer
    numStudentiCorso: array(range) of integer
    slotRichiestoCorso: array(range) of integer
    numero: integer
!definisco i dati del file che contiene i Laib
    numLaib: integer
    nomeLaib: array(range) of string
    numeroPC: array(range) of integer
    capienzaLaib: array(range) of integer
!definisco i dati del file che contiene gli slot orari
    numSlot: integer
    nomeSlot: array(range) of integer
    nomeGiorno: array(range) of string
    tipoSlot: array(range) of string
!definisco i dati del file dei Corsi con Orario di Riserva
    numCorsiRiserva: integer
    numeroRiserva: integer
    nomeCorsoRiserva: array(range) of string
    pesoCorsoRiserva: array(range) of integer
    numStudentiCorsoRiserva: array(range) of integer
    slotRichiestoCorsoRiserva: array(range) of integer
end-declarations
!lettura dei file di ingresso (inizializzazione dei dati)

fopen(fileCorsi, F_INPUT)
readln(numCorsi)
forall(cnt in 1..numCorsi) do
    readln(nomeCorso(cnt), numero,
pesoCorso(cnt), numStudentiCorso(cnt), slotRichiestoCorso(cnt))
end-do

```

```

fclose(F_INPUT)

fopen(fileLaib,F_INPUT)
readln(numLaib)
forall(cnt in 1..numLaib) do
    readln(nomeLaib(cnt),numero, numeroPC(cnt), capienzaLaib(cnt))
end-do
fclose(F_INPUT)

fopen(fileSlot,F_INPUT)
readln(numSlot)
forall(cnt in 1..numSlot) do
    readln(nomeSlot(cnt),nomeGiorno(cnt),tipoSlot(cnt))
end-do
fclose(F_INPUT)

!lettura del file CorsiRiserva
fopen(fileCorsiRiserva,F_INPUT)
readln(numCorsiRiserva)
forall(cnt in 1..numCorsiRiserva) do
    readln(nomeCorsoRiserva(cnt), numeroRiserva,
pesoCorsoRiserva(cnt),numStudentiCorsoRiserva(cnt),
    slotRichiestoCorsoRiserva(cnt))
end-do
fclose(F_INPUT)

! how to print to screen
!forall(cnt in 1..numLaib) do
!    writeln(nomeLaib(cnt)," ",numero," ",numeroPC(cnt)," ", capienzaLaib(cnt))
!end-do

! stampa i dati degli slot
!forall(cnt in 1..numSlot) do
!    writeln(nomeSlot(cnt)," ",nomeGiorno(cnt)," ",tipoSlot(cnt))
!end-do

! how to print to screen
!forall(cnt in 1..numCorsiRiserva) do
!    writeln(nomeCorsoRiserva(cnt)," ",numStudentiCorsoRiserva(cnt)," ",
slotRichiestoCorsoRiserva(cnt))
!end-do

!definisco la variabile decisionale x(i,j,k): se assegno il corso i al laib j
nello slot richiesto k
!scrivendo in ordine definisco la i la j e il k
declarations
    x: array(1..numCorsi,1..numLaib,1..numSlot) of mpvar
!definisco una variabile "ausiliaria": se assegno o meno un corso
    y: array(1..numCorsi) of mpvar
end-declarations
!che sono binarie
forall( i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot) x(i,j,k) is_binary

forall(i in 1..numCorsi) y(i) is_binary !oppure scrivo y(i) <= 1

!corrispondenza tra la variabile decisionale x e la variabile ausiliaria y:
forall(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot) x(i,j,k)<=y(i)

! Ogni corso PUO' (metto quindi il <=1 ) essere assegnato ad un laib e in uno
slot
forall(i in 1..numCorsi) sum(j in 1..numLaib, k in 1..numSlot) x(i,j,k)<= 1
!SE METTO x(i,j,k)=1 allora DEVE essere assegnato

```

```

!Per ogni laib e per ogni slot ci può essere al più un corso
forall(j in 1..numLaib, k in 1..numSlot) sum(i in 1..numCorsi) x(i,j,k) <= 1

!Per il LAIB1: ci possono stare due corsi al laib, ma rispettando la capienza
massima
forall(k in 1..numSlot) ( (sum(i in 1..numCorsi)x(i,1,k)+ sum(i in
1..numCorsi)x(i,2,k)) <=
2*(1-(sum(i in 1..numCorsi)x(i,3,k))) )

!Per ogni laib e per ogni slot orario il corso non deve superare la capienza
massima
forall(j in 1..numLaib, k in 1..numSlot) sum(i in 1..numCorsi)
numStudentiCorso(i)*x(i,j,k)<=
capienzaLaib(j)

!I corsi da 3 slot devono essere negli Slot richiesti e consecutivi
forall(i in 1..numCorsida3 | (i+2) mod 3 = 0) do
    sum(j in 1..numLaib) (x(i,j,slotRichiestoCorso(i)) +
x(i+1,j,slotRichiestoCorso(i+1)) +
    x(i+2,j,slotRichiestoCorso(i+2))) = (y(i) + y(i+1) + y(i+2))
end-do

!I corsi da 3 slot devono essere negli slot richiesti, consecutivi e NELLO
STESSO LAIB
forall(i in 1..numCorsida3, j in 1..numLaib) do
    if((i+2) mod 3 = 0) then
        x(i,j,slotRichiestoCorso(i)) = x(i+1,j,slotRichiestoCorso(i+1))
        x(i+1,j,slotRichiestoCorso(i+1)) = x(i+2,j,slotRichiestoCorso(i+2))
    end-if
end-do

!I corsi da 2 slot devono essere negli Slot richiesti e consecutivi
forall(i in numCorsida3+1..numCorsida2 | ((i-numCorsida3)+1) mod 2 = 0) do
    sum(j in 1..numLaib) (x(i,j,slotRichiestoCorso(i)) +
x(i+1,j,slotRichiestoCorso(i+1))) =
    (y(i) + y(i+1))
end-do

!I corsi da 2 slot devono essere negli slot richiesti, consecutivi e nello
stesso laib
forall(i in numCorsida3+1..numCorsida2, j in 1..numLaib) do
    if(((i-numCorsida3)+1) mod 2 = 0) then
        x(i,j,slotRichiestoCorso(i)) = x(i+1,j,slotRichiestoCorso(i+1))
    end-if
end-do

!I corsi da un solo slot:
forall(i in numCorsida2+1..numCorsi) do !| (i) mod 1 = 0
    sum(j in 1..numLaib) (x(i,j,slotRichiestoCorso(i))) = (y(i))
end-do

!C'è un corso che utilizza attrezzature al momento disponibili solo al LAIB5 e
quindi
!devo assegnare quel corso al LAIB5 negli slot richiesti:
x(127,6,18)=1
x(128,6,19)=1
x(129,6,21)=1
x(130,6,22)=1

!C'è un corso che utilizza un software al momento disponibile solo al LAIB5 e
quindi
!devo assegnare quel corso al LAIB5 negli slot richiesti:
x(121,6,8)=1

```

```

x(122,6,9)=1

!Definisco la funzione obiettivo
!obj:= sum(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot)
pesoCorso(i)*x(i,j,k)
!Visto che uso la variabile ausiliaria y, la funzione obiettivo diventa:
obj1:= sum(i in 1..numCorsi) pesoCorso(i)*y(i)
!Devo massimizzare il numero di corsi i che posso assegnare al laib j nello slot
richiesto k
maximize(obj1)
!Definisco il risultato: la somma dei corsi i che riesco ad assegnare al laib j
ris:= sum(i in 1..numCorsi) y(i)

! ---- PARTE che visualizzo nella schermata di Output/Input ----
writeln("Il numero dei corsi che riesco ad assegnare al laib sono:
",getsol(ris)," su ",
numCorsi)

writeln("\n","I corsi che vengono assegnati sono", "\n")

!Stampa su schermata i corsi i che vengono assegnati al laib j nello slot
richiesto k
forall(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot) do
    if(getsol(x(i,j,k)) >0.5) then
        writeln(nomeCorso(i), " ", numStudentiCorso(i), "\t", nomeGiorno(k), " ",
tipoSlot(k), "\t",
        nomeLaib(j))
    end-if
end-do
!Stampa su schermata qual è la soluzione ottimale
writeln("\n","Best solution: ", "\n", getobjval )
!oppure posso scrivere
!writeln("\n","Best solution: ", "\n", getsol(obj1) )

writeln("\n","Il numero dei corsi che NON riesco ad assegnare al laib sono:
",getsol(numCorsi-ris))

writeln("\n","I corsi NON assegnati sono:", "\n")

!Stampa su schermata i corsi che non sono assegnati
forall(i in 1..numCorsi, k in 1..numSlot | k=slotRichiestoCorso(i)) do
    if(getsol(y(i))=0 ) then
        writeln(nomeCorso(i), "\t", numStudentiCorso(i), "\t", nomeGiorno(k), " ",
tipoSlot(k))
    end-if
end-do

!Stampa su schermata quali sono i laib e gli slot orari rimasti liberi?
writeln("\n","I laib e gli slot orari rimasti liberi sono:", "\n")
!Se sono liberi il LAIB1a e il LAIB1b e il LAIB1 allora stampa solo il LAIB1
forall(k in 1..numSlot) do
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
(sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
(sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        !writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
        !writeln(nomeLaib(2), "\t", nomeGiorno(k), " ", tipoSlot(k))
        writeln(nomeLaib(3), "\t", nomeGiorno(k), " ", tipoSlot(k))
    else !altrimenti
!Se sono liberi il LAIB1a e il LAIB1 allora stampa solo il LAIB1a
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
(sum(i in 1..numCorsi) getsol(x(i,2,k)) > 0.1) and
(sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
    end-if
end-if
end-do

```

```

        else
!Se sono liberi il LAIB1b e il LAIB1 allora stampa solo il LAIB1b
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) > 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        !writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
        writeln(nomeLaib(2), "\t", nomeGiorno(k), " ", tipoSlot(k))
    end-if
end-if
end-if
end-do

!potrei scriverlo anche così:
!forall(j in 1..3,k in 1..numSlot) do
!    if((sum(i in 1..numCorsi) getsol(x(i,j,k)) < 0.1)) then
!        writeln(nomeLaib(j), " ", nomeGiorno(k), " ", tipoSlot(k))
!    end-if
!end-do

!Per gli altri laib, esclusi il LAIB1, LAIB1a e il LAIB1b
forall(j in 1..numLaib,k in 1..numSlot | j <> 3 and j <> 1 and j <> 2) do
    if (sum (i in 1..numCorsi) getsol(x(i,j,k))=0)then
        writeln(nomeLaib(j), "\t", nomeGiorno(k), " ", tipoSlot(k))
    end-if
end-do

!potrei scriverlo anche così:
!forall(j in 4..numLaib,k in 1..numSlot) do
!    if (sum (i in 1..numCorsi) getsol(x(i,j,k))=0)then
!        writeln(nomeLaib(j), " ", nomeGiorno(k), " ", tipoSlot(k))
!    end-if
!end-do

!---- Per SALVARE in un file (di testo) il risultato che vedo sullo schermo di
Output/Input ----

fopen(outrisultato, F_OUTPUT)

writeln("Il numero dei corsi che riesco ad assegnare al laib sono:
",getsol(ris)," su ",numCorsi)

writeln("\n","I corsi che vengono assegnati sono", "\n")

forall(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot) do
    if(getsol(x(i,j,k)) >0.5) then
        writeln(nomeCorso(i), " ", numStudentiCorso(i),"\t", nomeGiorno(k), "
",tipoSlot(k),
        "\t", nomeLaib(j))
    end-if
end-do
writeln("\n","Best solution: ", "\n", getobjval )

writeln("\n","Il numero dei corsi che NON riesco ad assegnare al laib sono:
",getsol(numCorsi-ris))

writeln("\n","I corsi NON assegnati sono:", "\n")

forall(i in 1..numCorsi, k in 1..numSlot | k=slotRichiestoCorso(i)) do
    if(getsol(y(i))=0 ) then
writeln(nomeCorso(i), " ", numStudentiCorso(i), " ", nomeGiorno(k), " ",
tipoSlot(k))
    end-if
end-do

```

```

writeln("\n","I laib e gli slot orari rimasti liberi sono:", "\n")
!Se sono liberi il LAIB1a e il LAIB1b e il LAIB1 allora stampa solo il LAIB1
forall(k in 1..numSlot) do
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        !writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
        !writeln(nomeLaib(2), "\t", nomeGiorno(k), " ", tipoSlot(k))
        writeln(nomeLaib(3), "\t", nomeGiorno(k), " ", tipoSlot(k))
    else !altrimenti
!Se sono liberi il LAIB1a e il LAIB1 allora stampa solo il LAIB1a
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) > 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
    else
!Se sono liberi il LAIB1b e il LAIB1 allora stampa solo il LAIB1b
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) > 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        !writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
        writeln(nomeLaib(2), "\t", nomeGiorno(k), " ", tipoSlot(k))
    end-if
    end-if
    end-if
end-do

!Per gli altri laib, esclusi il LAIB1, LAIB1a e il LAIB1b
forall(j in 1..numLaib,k in 1..numSlot | j <> 3 and j <> 1 and j <> 2) do
    if (sum (i in 1..numCorsi) getsol(x(i,j,k))=0)then
        writeln(nomeLaib(j), "\t", nomeGiorno(k), " ", tipoSlot(k))
    end-if
end-do

fclose(F_OUTPUT)

!---- Per SALVARE in un file (excel) I CORSI ASSEGNATI ----
fopen(outCorsiAssegnati, F_OUTPUT)
forall(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot) do
    if(getsol(x(i,j,k)) >0.5) then
        writeln(nomeCorso(i), "\t", numStudentiCorso(i),"\t",
nomeSlot(k),"\t",nomeGiorno(k),
"\t", tipoSlot(k),"\t", nomeLaib(j))
    end-if
end-do
fclose(F_OUTPUT)

!---- Per SALVARE in un file (di testo) I CORSI NON ASSEGNATI ----
fopen(outCorsiNonAssegnati, F_OUTPUT)

writeln(getsol(numCorsi-ris))

forall(i in 1..numCorsi, k in 1..numSlot | k=slotRichiestoCorso(i)) do
    if(getsol(y(i))=0 ) then
        writeln(nomeCorso(i)," ",i," ",pesoCorso(i)," ", numStudentiCorso(i), " ",
nomeSlot(k))
    end-if
end-do

fclose(F_OUTPUT)

!---- Per SALVARE in un file (excel) I LAIB E GLI SLOT RIMASTI LIBERI ----

```

```

fopen(outLaibrimasti, F_OUTPUT)

!Se sono liberi il LAIB1a e il LAIB1b e il LAIB1 allora stampa solo il LAIB1
forall(k in 1..numSlot) do
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        !writeln(nomeLaib(1), "\t", nomeGiorno(k), " ", tipoSlot(k))
        !writeln(nomeLaib(2), "\t", nomeGiorno(k), " ", tipoSlot(k))
        writeln(nomeLaib(3), "\t", nomeSlot(k), "\t", nomeGiorno(k), "\t",
tipoSlot(k))
    else !altrimenti
!Se sono liberi il LAIB1a e il LAIB1 allora stampa solo il LAIB1a
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) > 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        writeln(nomeLaib(1), "\t", nomeSlot(k), "\t", nomeGiorno(k), "\t",
tipoSlot(k))
    else
!Se sono liberi il LAIB1b e il LAIB1 allora stampa solo il LAIB1b
    if((sum(i in 1..numCorsi) getsol(x(i,1,k)) > 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,2,k)) < 0.1) and
        (sum(i in 1..numCorsi) getsol(x(i,3,k)) < 0.1) ) then
        writeln(nomeLaib(2), "\t", nomeSlot(k), "\t", nomeGiorno(k), "\t",
tipoSlot(k))
    end-if
    end-if
    end-if
end-do

!Per gli altri laib, esclusi il LAIB1, LAIB1a e il LAIB1b
forall(j in 1..numLaib, k in 1..numSlot | j <> 3 and j <> 1 and j <> 2) do
    if (sum (i in 1..numCorsi) getsol(x(i,j,k))=0) then
        writeln(nomeLaib(j), "\t", nomeSlot(k), "\t", nomeGiorno(k), "\t",
tipoSlot(k))
    end-if
end-do

fclose(F_OUTPUT)

!PARTE PER I CORSI CHE NON SONO RIUSCITA AD ASSEGNARE E CHE ORA RIPROVO CON
ORARIO DI RISERVA:
!variabile decisionale p(i,j,k); scrivendo in ordine definisco la i la j e il k
declarations
    p: array(1..numCorsiRiserva,1..numLaib,1..numSlot) of mpvar
!definisco una variabile "ausiliaria": se assegno o meno un corso
    q: array(1..numCorsiRiserva) of mpvar
end-declarations
!che sono binarie
forall( i in 1..numCorsiRiserva, j in 1..numLaib, k in 1..numSlot) p(i,j,k)
is_binary

forall(i in 1..numCorsiRiserva) q(i) is_binary !oppure scrivo q(i) <= 1

!corrispondenza tra la variabile decisionale p e la variabile ausiliaria q:
forall(i in 1..numCorsiRiserva, j in 1..numLaib, k in 1..numSlot) p(i,j,k)<=q(i)

! Ogni corso PUO' (metto quindi il <=1 ) essere assegnato ad un laib e in uno
slot
forall(i in 1..numCorsiRiserva) sum(j in 1..numLaib, k in 1..numSlot) p(i,j,k)<=
1

```

```

!Per ogni laib e per ogni slot ci può essere al più un corso
forall(j in 1..numLaib, k in 1..numSlot) sum(i in 1..numCorsiRiserva) p(i,j,k)
<= 1

!Per il LAIB1: ci possono stare due corsi al laib, ma rispettando la capienza
forall(k in 1..numSlot) ((sum(i in 1..numCorsiRiserva)p(i,1,k)+ sum(i in
1..numCorsiRiserva)p(i,2,k))<=
    2*(1- sum(i in 1..numCorsiRiserva)p(i,3,k) ))

!Per ogni laib e per ogni slot orario il corso non deve superare la capienza
massima
forall(j in 1..numLaib, k in 1..numSlot) (sum(i in 1..numCorsiRiserva)
numStudentiCorsoRiserva(i)*p(i,j,k)<= capienzaLaib(j))

!Le 3 squadre dei corsi devono essere negli Slot richiesti
forall(i in 1..numCorsida3Riserva | (i+2) mod 3 = 0) do
    sum(j in 1..numLaib) ( p(i,j,slotRichiestoCorsoRiserva(i)) +
        p(i+1,j,slotRichiestoCorsoRiserva(i+1)) +
        p(i+2,j,slotRichiestoCorsoRiserva(i+2)) ) = (q(i) + q(i+1) + q(i+2))
end-do

!tutte le 3 squadre dei Corsi devono essere negli slot richiesti e NELLO STESSO
LAIB
forall(i in 1..numCorsida3Riserva, j in 1..numLaib) do
    if((i+2) mod 3 = 0) then
        p(i,j,slotRichiestoCorsoRiserva(i)) =
p(i+1,j,slotRichiestoCorsoRiserva(i+1))
        p(i+1,j,slotRichiestoCorsoRiserva(i+1)) =
p(i+2,j,slotRichiestoCorsoRiserva(i+2))
    end-if
end-do

!Le 2 squadre dei corsi devono essere negli Slot richiesti
forall(i in numCorsida3Riserva+1..numCorsida2Riserva | ((i-
numCorsida3Riserva)+1) mod 2 = 0) do
    sum(j in 1..numLaib) (p(i,j,slotRichiestoCorsoRiserva(i)) +
        p(i+1,j,slotRichiestoCorsoRiserva(i+1))) = (q(i) +
q(i+1))
end-do

!tutte le 2 squadre dei Corsi devono essere negli slot richiesti e NELLO STESSO
LAIB
forall(i in numCorsida3Riserva+1..numCorsida2Riserva, j in 1..numLaib) do
    if(((i-numCorsida3Riserva)+1) mod 2 = 0) then
        p(i,j,slotRichiestoCorsoRiserva(i)) =
p(i+1,j,slotRichiestoCorsoRiserva(i+1))
    end-if
end-do

!i corsi da un solo slot:
forall(i in numCorsida2Riserva+1..numCorsiRiserva ) do !| (i) mod 1 = 0
    sum(j in 1..numLaib) (p(i,j,slotRichiestoCorsoRiserva(i))) = (q(i))
end-do

!funzione obiettivo: massimizza il numero di corsi
!obj:= sum(i in 1..numCorsi, j in 1..numLaib, k in 1..numSlot)
pesoCorso(i)*x(i,j,k)
obj3:= sum(i in 1..numCorsiRiserva) pesoCorsoRiserva(i)*q(i)
maximize(obj3)

ris2:= sum(i in 1..numCorsiRiserva) q(i)

```

```

writeln("\n","Il numero dei corsi che riesco a riassegnare al laib sono:
",getsol(ris2)," su ",
numCorsiRiserva)

writeln("\n","I corsi che vengono riassegnati sono", "\n")

!così vedo i corsi che vengono riassegnati al laib
forall(i in 1..numCorsiRiserva, j in 1..numLaib, k in 1..numSlot) do
    if(getsol(p(i,j,k)) >0.5) then
        writeln(nomeCorsoRiserva(i), "\t", numStudentiCorsoRiserva(i), "\t",
nomeLaib(j), " ",
        nomeGiorno(k), " ", tipoSlot(k))
    end-if
end-do

writeln("\n","Il numero dei corsi che NON riesco riassegnare al laib sono: ",
    getsol(numCorsiRiserva-ris2))

!Per SALVARE LA PARTE FINALE DEI CORSI RIASSEGNATI CON L'ORARIO DI RISERVA

fopen(outfileRiserva, F_OUTPUT)
writeln("\n","I corsi che vengono riassegnati sono", "\n")

!così vedo i corsi che vengono riassegnati al laib
forall(i in 1..numCorsiRiserva, j in 1..numLaib, k in 1..numSlot) do
    if(getsol(p(i,j,k)) >0.5) then
        writeln(nomeCorsoRiserva(i), " ", nomeLaib(j), " ", nomeGiorno(k), " ",
tipoSlot(k))
    end-if
end-do

!writeln("\n","Best solution: ", "\n", getobjval )
!oppure posso scrivere
!writeln("\n","Best solution: ", "\n", getsol(obj2) )

writeln("\n","Il numero dei corsi che riesco a riassegnare al laib sono:
",getsol(ris2)," su ",
numCorsiRiserva)

writeln("\n","Il numero dei corsi che NON riesco riassegnare al laib sono: ",
    getsol(numCorsiRiserva-ris2))
fclose(F_OUTPUT)

end-model

```