

CR TP4 Programmation par contraintes

Les régates

Julien LETOILE, Romain HUBERT

le 08/03/2021

Table des matières

Table des matières

I. Réponses rédigées

Question 4.1

Question 4.2

Question 4.3

Question 4.4

Question 4.5

Question 4.6

Question 4.7

Question 4.8

II Anexes

Code source

I. Réponses rédigées

Question 4.1

```
1  getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,  
    NbConf):-  
2      TailleEquipes = [(7, 6, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 3,  
    3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2),  
3      NbEquipes  #= 29,  
4      CapaBateaux = [(10, 10, 9, 8, 8, 8, 8, 8, 8, 7, 6, 4, 4),  
5      NbBateaux  #= 13,  
6      NbConf    #= 7.
```

Test:

```
getData(T, N, C, Nb, NbConf).
```

```
T = [(5, 5, 2, 1)
```

```
N = 4
```

```
C = [(7, 6, 5)
```

```
Nb = 3
```

```
NbConf = 3
```

Question 4.2

```
1  defineVars(T, NbEquipes, NbConf, NbBateaux):-  
2      dim(T, [NbEquipes, NbConf]),  
3      T :: 1..NbBateaux.
```

Test:

```
solve(T).
```

```
T = [(423{1 .. 3}, _438{1 .. 3}, _453{1 .. 3}), [(468{1 .. 3}, 483{1 .. 3}, _498{1 .. 3}), [(513{1 .. 3}, 528{1 .. 3}, _543{1 .. 3}), [(558{1 .. 3}, _573{1 .. 3}, _588{1 .. 3})]
```

Question 4.3

```
1  getVarList(T, L):-  
2      term_variables(T, L).
```

Question 4.4

```
1  solve(T):-  
2      getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,  
      NbConf),  
3      defineVars(T, NbEquipes, NbConf, NbBateaux),  
4      pasMemeBateaux(T, NbEquipes, NbConf),  
5      pasMemePartenaires(T, NbEquipes, NbConf),  
6      capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux,  
      NbBateaux, NbConf),  
7      getVarListAlt(T, Liste),  
8      labeling(Liste),  
9      affiche(T). /* Ajouté à la fin du TP */
```

Test:

```
solve(T).
```

```
T = [(1, 1, 1), (1, 1, 1), (1, 1, 1), (1, 1, 1)]  
Yes (0.00s cpu, solution 1, maybe more) ? ;
```

```
T = [(1, 1, 1), (1, 1, 1), (1, 1, 1), (1, 1, 2)]  
Yes (0.00s cpu, solution 2, maybe more) ? ;
```

```
T = [(1, 1, 1), (1, 1, 1), (1, 1, 1), (1, 1, 3)]  
Yes (0.00s cpu, solution 3, maybe more) ?
```

Question 4.5

```
1  pasMemeBateaux(T, NbEquipes, NbConf):-
2      (
3          /* Pour chaque Equipe */
4          for(Equipe, 1, NbEquipes),
5          param(T, NbConf)
6      do
7          (
8              /* Pour chaque Confrontation */
9              for(I, 1, NbConf),
10             param(T, Equipe, NbConf)
11         do
12             (
13                 /* Pour chaque confrontation suivante */
14                 for(J, I+1, NbConf),
15                 param(T),
16                 param(I),
17                 param(Equipe)
18             do
19                 /*
20                  On pose la contrainte qu'une équipe, à une
21                  confrontation donné, n'aura pas le même bateau dans les
22                  confrontations futures
23                  */
24                 X is T[Equipe, I],
25                 Y is T[Equipe, J],
26                 X #\= Y
27             )
28         )
29     ).
```

Test:

T = [[(1, 2, 3), (1, 2, 3), (1, 2, 3), (1, 2, 3)]
Yes (0.00s cpu, solution 1, maybe more) ? ;

T = [[(1, 2, 3), (1, 2, 3), (1, 2, 3), (1, 3, 2)]
Yes (0.00s cpu, solution 2, maybe more) ? ;

T = [(1, 2, 3), (1, 2, 3), (1, 2, 3), (2, 1, 3)]

Yes (0.00s cpu, solution 3, maybe more) ? ;

Question 4.6

```
1  pasMemePartenaires(T, NbEquipes, NbConf):-
2      (
3          for(Equipe1, 1, NbEquipes),
4          param(T, NbEquipes, NbConf)
5      do
6          (
7              for(Equipe2, Equipe1 + 1, NbEquipes),
8              param(T, Equipe1, NbConf)
9          do
10             (
11                 for(Conf1, 1, NbConf),
12                 param(T, Equipe1, Equipe2, NbConf)
13             do
14                 (
15                     for(Conf2, Conf1+1, NbConf),
16                     param(T, Equipe1, Equipe2, Conf1)
17                 do
18                     /* Equipes de la première confrontation */
19                     E1_C1 is T[Equipe1, Conf1],
20                     E2_C1 is T[Equipe2, Conf1],
21
22                     /* Equipes de la deuxième confrontation */
23                     E1_C2 is T[Equipe1, Conf2],
24                     E2_C2 is T[Equipe2, Conf2],
25
26                     /* Si deux équipes sont ensemble dans une
27 confrontation alors elles ne le sont pas lors de l'autre
28 confrontation */
29
30                     (E1_C1 #= E2_C1) => (E1_C2 #\= E2_C2),
31                     (E1_C2 #= E2_C2) => (E1_C1 #\= E2_C1)
32                 )
33             )
34         )
35     ).
```

Test:

```
solve(T)
```

```
T = [(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1)]
```

```
Yes (0.00s cpu, solution 1, maybe more) ? ;
```

```
T = [(1, 2, 3), (1, 3, 2), (2, 1, 3), (3, 1, 2)]
```

```
Yes (0.00s cpu, solution 2, maybe more) ? ;
```

```
T = [(1, 2, 3), (1, 3, 2), (2, 1, 3), (3, 2, 1)]
```

```
Yes (0.00s cpu, solution 3, maybe more) ? ;
```

```
T = [(1, 2, 3), (1, 3, 2), (2, 3, 1), (2, 1, 3)]
```

```
Yes (0.00s cpu, solution 4, maybe more) ?
```

Question 4.7

```
1  capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
   NbConf):-
2      (
3          /* Pour chaque confrontation */
4          for(NumConf, 1, NbConf),
5          param(T, TailleEquipes, NbEquipes, CapaBateaux,
   NbBateaux)
6      do
7          (
8              for(NumBateau, 1, NbBateaux),
9              param(T, TailleEquipes, NbEquipes, CapaBateaux,
   NumConf)
10         do
11             /* On impose que Capa est inférieur à la capacité
   maximale du bateau */
12             Capa #=< CapaBateaux[NumBateau],
13             (
14                 for(NumEquipe, 1, NbEquipes),
15                 /* Accumuler (faire un +=), le résultat final
   est dans Capa */
```

```

16         fromto(0, C, NC, Capa),
17         param(T, TailleEquipes, NumBateau, NumConf)
18     do
19         NB is T[NumEquipe, NumConf],
20         NC #= C + TailleEquipes[NumEquipe] * (NB #=
           NumBateau)
21     )
22 )
23 ).

```

Test:

T = [(1, 2, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1))

Yes (0.00s cpu, solution 1, maybe more) ? ;

T = [(1, 2, 3), (3, 1, 2), (2, 3, 1), (1, 3, 2))

Yes (0.00s cpu, solution 2, maybe more) ? ;

T = [(1, 3, 2), (2, 1, 3), (3, 2, 1), (3, 1, 2))

Yes (0.00s cpu, solution 3, maybe more) ? ;

T = [(1, 3, 2), (3, 2, 1), (2, 1, 3), (1, 2, 3))

Yes (0.00s cpu, solution 4, maybe more) ? ;

T = [(2, 1, 3), (1, 3, 2), (3, 2, 1), (3, 1, 2))

Yes (0.00s cpu, solution 5, maybe more) ?

12 solutions en tout

Question 4.8

```

1  getVarListAlt(T, L):-
2      dim(T, [NbEquipes, NbConf]),
3      (
4          for(Conf, 1, NbConf),
5          fromto([], AncienneConf, NouvelleConf, L),
6          param(T, NbEquipes)
7      do

```



```

8      /* Prendre la moitié */
9      Moitie #= div(NbEquipes, 2) + 1,
10     (
11         for(IdGrandeEquipe, 1, Moitie),
12         fromto([], AncienneListe, NouvelleListe,
EquipesConf),
13         param(T, NbEquipes, Conf)
14     do
15         GrandeEquipe is T[IdGrandeEquipe, Conf],
16
17         /* Prendre la petite équipe du même indice que la
grande équipe, mais en partant de la fin */
18         IdPetiteEquipe #= NbEquipes - IdGrandeEquipe + 1,
19
20         /* Test si l'id de la petite équipe est plus grande
que l'autre */
21         /* Evite de le mettre en double si le nombre
d'équipes est impaire */
22         ( IdPetiteEquipe #> IdGrandeEquipe ->
23             PetiteEquipe is T[IdPetiteEquipe, Conf],
24             NouvelleListe = [GrandeEquipe, PetiteEquipe |
AncienneListe]
25         ;
26             NouvelleListe = [GrandeEquipe | AncienneListe]
27         )
28
29     ),
30     append(AncienneConf, EquipesConf, NouvelleConf)
31 ).

```

Test:

10	7	9	3	8	1	2
11	9	8	7	2	10	1
9	11	10	8	3	2	7
8	10	11	9	1	6	5
7	8	6	10	1	2	3
12	6	13	11	10	9	4
6	12	4	13	11	5	10
5	6	7	1	9	13	12

4	5	1	2	9	3	11
4	2	5	1	6	7	3
3	1	2	6	7	8	4
2	3	1	4	7	5	9
2	1	3	5	6	4	8
1	3	2	5	4	7	6
1	2	3	4	5	6	7
1	4	5	2	3	8	9
1	5	4	3	2	9	8
2	4	6	1	5	3	10
3	2	1	7	4	9	5
3	4	7	8	2	1	6
5	1	4	2	10	12	13
5	3	8	6	12	4	2
6	4	2	9	12	11	13
6	5	3	10	13	12	1
7	9	5	6	13	11	8
8	13	7	12	3	4	11
9	13	12	2	4	11	1
13	8	10	12	11	3	9
13	10	12	11	5	8	6

$T = \{ \pi(10, 7, 9, 3, 8, 1, 2), \pi(11, 9, 8, 7, 2, 10, 1), \pi(9, 11, 10, 8, 3, 2, 7), \pi(8, 10, 11, 9, 1, 6, 5), \pi(7, 8, 6, 10, 1, 2, 3), \pi(12, 6, 13, 11, 10, 9, 4), \pi(6, 12, 4, 13, 11, 5, 10), \pi(5, 6, 7, 1, 9, 13, 12), \pi(4, 5, 1, 2, 9, 3, 11), \pi(4, 2, 5, 1, 6, 7, 3), \pi(3, 1, 2, 6, 7, 8, 4), \pi(2, 3, 1, 4, 7, 5, 9), \pi(2, 1, 3, 5, 6, 4, 8), \pi(1, 3, 2, 5, 4, 7, 6), \pi(1, 2, 3, 4, 5, 6, 7), \pi(1, 4, 5, 2, 3, 8, 9), \pi(1, 5, 4, 3, 2, 9, 8), \pi(2, 4, 6, 1, 5, 3, 10), \pi(3, 2, 1, 7, 4, 9, 5), \pi(3, 4, 7, 8, 2, 1, 6), \pi(5, 1, 4, 2, 10, 12, 13), \pi(5, 3, 8, 6, 12, 4, 2), \pi(6, 4, 2, 9, 12, 11, 13), \pi(6, 5, 3, 10, 13, 12, 1), \pi(7, 9, 5, 6, 13, 11, 8), \pi(8, 13, 7, 12, 3, 4, 11), \pi(9, 13, 12, 2, 4, 11, 1), \pi(13, 8, 10, 12, 11, 3, 9), \pi(13, 10, 12, 11, 5, 8, 6) \}$

Yes (152.61s cpu, solution 1, maybe more) ?

II Anexes

Code source

```
1  :-lib(ic).
2  :-lib(ic_symbolic).
3
4  /* Question 4.4 */
5  solve(T):-
6      getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
7      NbConf),
8      defineVars(T, NbEquipes, NbConf, NbBateaux),
9      pasMemeBateaux(T, NbEquipes, NbConf),
10     pasMemePartenaires(T, NbEquipes, NbConf),
11     capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux,
12     NbBateaux, NbConf),
13     getVarListAlt(T, Liste),
14     labeling(Liste),
15     affiche(T).
16
17 /* Test 4.4 */
18
19 T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 1))
20 Yes (0.00s cpu, solution 1, maybe more) ? ;
21
22 T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 2))
23 Yes (0.00s cpu, solution 2, maybe more) ? ;
24
25 T = []([](1, 1, 1), [](1, 1, 1), [](1, 1, 1), [](1, 1, 3))
26 Yes (0.00s cpu, solution 3, maybe more) ?
27 */
28
29 /* Question 4.1 */
30 % getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
31 %      NbConf):-
32 %      TailleEquipes = [](5, 5, 2, 1),
```

```

32 %      NbEquipes #= 4,
33 %      CapaBateaux = [](7, 6, 5),
34 %      NbBateaux #= 3,
35 %      NbConf #= 3.
36
37 getData(TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
NbConf):-
38     TailleEquipes = [](7, 6, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4, 3,
3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2),
39     NbEquipes #= 29,
40     CapaBateaux = [](10, 10, 9, 8, 8, 8, 8, 8, 8, 7, 6, 4, 4),
41     NbBateaux #= 13,
42     NbConf #= 7.
43
44 /* Test
45
46 getData(T, N, C, Nb, NbConf).
47
48 T = [](5, 5, 2, 1)
49 N = 4
50 C = [](7, 6, 5)
51 Nb = 3
52 NbConf = 3
53
54 */
55
56 /* Question 4.2 */
57 defineVars(T, NbEquipes, NbConf, NbBateaux):-
58     dim(T, [NbEquipes, NbConf]),
59     T :: 1..NbBateaux.
60
61 /*
62 T = []([][_423{1 .. 3}, _438{1 .. 3}, _453{1 .. 3}), [][_468{1
.. 3}, _483{1 .. 3}, _498{1 .. 3}), [][_513{1 .. 3}, _528{1 ..
3}, _543{1 .. 3}), [][_558{1 .. 3}, _573{1 .. 3}, _588{1 .. 3}))
63 */
64
65 /* Question 4.3 */
66 getVarList(T, L):-
67     term_variables(T, L).
68

```

```

69
70  /* Q 4.5 */
71  pasMemeBateaux(T, NbEquipes, NbConf):-
72      (
73          /* Pour chaque Equipe */
74          for(Equipe, 1, NbEquipes),
75          param(T, NbConf)
76      do
77          (
78              /* Pour chaque Confrontation */
79              for(I, 1, NbConf),
80              param(T, Equipe, NbConf)
81          do
82              (
83                  /* Pour chaque confrontation suivante */
84                  for(J, I+1, NbConf),
85                  param(T),
86                  param(I),
87                  param(Equipe)
88              do
89                  /*
90                     On pose la contrainte qu'une équipe, à une
91                     confrontation donné, n'aura pas le même bateau dans les
92                     confrontations futures
93                     */
94                  X is T[Equipe, I],
95                  Y is T[Equipe, J],
96                  X #\= Y
97              )
98          )
99      ).
100
101  /*
102  T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 2, 3))
103  Yes (0.00s cpu, solution 1, maybe more) ? ;
104
105  T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](1, 3, 2))
106  Yes (0.00s cpu, solution 2, maybe more) ? ;
107
108  T = []([](1, 2, 3), [](1, 2, 3), [](1, 2, 3), [](2, 1, 3))
109  Yes (0.00s cpu, solution 3, maybe more) ? ;

```

```

08  */
09
10
11  /* Prendre 2 equipes & 2 confrontations
12  /* Question 4.6 */
13
14  pasMemePartenaires(T, NbEquipes, NbConf):-
15      (
16          for(Equipe1, 1, NbEquipes),
17          param(T, NbEquipes, NbConf)
18      do
19          (
20              for(Equipe2, Equipe1 + 1, NbEquipes),
21              param(T, Equipe1, NbConf)
22          do
23              (
24                  for(Conf1, 1, NbConf),
25                  param(T, Equipe1, Equipe2, NbConf)
26              do
27                  (
28                      for(Conf2, Conf1+1, NbConf),
29                      param(T, Equipe1, Equipe2, Conf1)
30                  do
31                      /* Equipes de la première confrontation */
32                      E1_C1 is T[Equipe1, Conf1],
33                      E2_C1 is T[Equipe2, Conf1],
34
35                      /* Equipes de la deuxième confrontation */
36                      E1_C2 is T[Equipe1, Conf2],
37                      E2_C2 is T[Equipe2, Conf2],
38
39                      /* Si deux équipes sont ensemble dans une
40 confrontation alors elles ne le sont pas lors de l'autre
41 confrontation */
42                      (E1_C1 #= E2_C1) => (E1_C2 #\= E2_C2),
43                      (E1_C2 #= E2_C2) => (E1_C1 #\= E2_C1)
44                  )
45              )
46          )
47      ).

```

```

47  /*
48  solve(T)
49
50  T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](2, 3, 1))
51  Yes (0.00s cpu, solution 1, maybe more) ? ;
52
53  T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](3, 1, 2))
54  Yes (0.00s cpu, solution 2, maybe more) ? ;
55
56  T = []([](1, 2, 3), [](1, 3, 2), [](2, 1, 3), [](3, 2, 1))
57  Yes (0.00s cpu, solution 3, maybe more) ? ;
58
59  T = []([](1, 2, 3), [](1, 3, 2), [](2, 3, 1), [](2, 1, 3))
60  Yes (0.00s cpu, solution 4, maybe more) ?
61
62  */
63
64
65  /* Question 4.7 */
66
67
68  capaBateaux(T, TailleEquipes, NbEquipes, CapaBateaux, NbBateaux,
NbConf):-
69      (
70          /* Pour chaque confrontation */
71          for(NumConf, 1, NbConf),
72          param(T, TailleEquipes, NbEquipes, CapaBateaux,
NbBateaux)
73      do
74          (
75              for(NumBateau, 1, NbBateaux),
76              param(T, TailleEquipes, NbEquipes, CapaBateaux,
NumConf)
77          do
78              /* On impose que Capa est inférieur à la capacité
maximale du bateau */
79              Capa #=< CapaBateaux[NumBateau],
80              (
81                  for(NumEquipe, 1, NbEquipes),
82                  /* Accumuler (faire un +=), le résultat final
est dans Capa */

```

```

83         fromto(0, C, NC, Capa),
84         param(T, TailleEquipes, NumBateau, NumConf)
85     do
86         NB is T[NumEquipe, NumConf],
87         NC #= C + TailleEquipes[NumEquipe] * (NB #=
            NumBateau)
88     )
89 )
90 ).
91
92 /*
93
94 T = []([](1, 2, 3), [](2, 3, 1), [](3, 1, 2), [](3, 2, 1))
95 Yes (0.00s cpu, solution 1, maybe more) ? ;
96
97 T = []([](1, 2, 3), [](3, 1, 2), [](2, 3, 1), [](1, 3, 2))
98 Yes (0.00s cpu, solution 2, maybe more) ? ;
99
00 T = []([](1, 3, 2), [](2, 1, 3), [](3, 2, 1), [](3, 1, 2))
01 Yes (0.00s cpu, solution 3, maybe more) ? ;
02
03 T = []([](1, 3, 2), [](3, 2, 1), [](2, 1, 3), [](1, 2, 3))
04 Yes (0.00s cpu, solution 4, maybe more) ? ;
05
06 T = []([](2, 1, 3), [](1, 3, 2), [](3, 2, 1), [](3, 1, 2))
07 Yes (0.00s cpu, solution 5, maybe more) ?
08
09 12 solutions en tout
10 */
11
12
13 /* Question 4.8 */
14 getVarListAlt(T, L):-
15     dim(T, [NbEquipes, NbConf]),
16     (
17         for(Conf, 1, NbConf),
18         fromto([], AncienneConf, NouvelleConf, L),
19         param(T, NbEquipes)
20     do
21         /* Prendre la moitié */
22         Moitie #= div(NbEquipes, 2) + 1,

```



```

23      (
24          for(IdGrandeEquipe, 1, Moitie),
25          fromto([], AncienneListe, NouvelleListe,
EquipesConf),
26          param(T, NbEquipes, Conf)
27      do
28          GrandeEquipe is T[IdGrandeEquipe, Conf],
29
30          /* Prendre la petite équipe du même indice que la
grande équipe, mais en partant de la fin */
31          IdPetiteEquipe #= NbEquipes - IdGrandeEquipe + 1,
32
33          /* Test si l'id de la petite équipe est plus grande
que l'autre */
34          /* Evite de le mettre en double si le nombre
d'équipes est impaire */
35          ( IdPetiteEquipe #> IdGrandeEquipe ->
36              PetiteEquipe is T[IdPetiteEquipe, Conf],
37              NouvelleListe = [GrandeEquipe, PetiteEquipe |
AncienneListe]
38          ;
39              NouvelleListe = [GrandeEquipe | AncienneListe]
40          )
41
42      ),
43      append(AncienneConf, EquipesConf, NouvelleConf)
44  ).
45
46  /*
47  T = []([](10, 7, 9, 3, 8, 1, 2), [](11, 9, 8, 7, 2, 10, 1), []
(9, 11, 10, 8, 3, 2, 7), [](8, 10, 11, 9, 1, 6, 5), [](7, 8, 6,
10, 1, 2, 3), [](12, 6, 13, 11, 10, 9, 4), [](6, 12, 4, 13, 11,
5, 10), [](5, 6, 7, 1, 9, 13, 12), [](4, 5, 1, 2, 9, 3, 11), []
(4, 2, 5, 1, 6, 7, 3), [](3, 1, 2, 6, 7, 8, 4), [](2, 3, 1, 4,
7, 5, 9), [](2, 1, 3, 5, 6, 4, 8), [](1, 3, 2, 5, 4, 7, 6), []
(1, 2, 3, 4, 5, 6, 7), [](1, 4, 5, 2, 3, 8, 9), [](1, 5, 4,
48  3, 2, 9, 8), [](2, 4, 6, 1, 5, 3, 10), [](3, 2, 1, 7, 4, 9, 5),
[](3, 4, 7, 8, 2, 1, 6), [](5, 1, 4, 2, 10, 12, 13), [](5, 3, 8,
6, 12, 4, 2), [](6, 4, 2, 9, 12, 11, 13), [](6, 5, 3, 10, 13,
12, 1), [](7, 9, 5, 6, 13, 11, 8), [](8, 13, 7, 12, 3, 4, 11),
[](9, 13, 12, 2, 4, 11, 1), [](13, 8,

```

```
49 10, 12, 11, 3, 9), [])(13, 10, 12, 11, 5, 8, 6))
50
51 Yes (152.61s cpu, solution 1, maybe more) ?
52 */
53
54 affiche(T):-
55     dim(T, [NbEquipes, NbConf]),
56     (
57         for(Equipe, 1, NbEquipes),
58         param(T, NbConf)
59     do
60         (
61             for(Conf, 1, NbConf),
62             param(T, Equipe)
63         do
64             X is T[Equipe, Conf],
65             write(X),
66             write("\t")
67         ),
68         writeln("")
69     )
```