Tricot en rond:
répartition de *d* diminutions sur un tour de *n* mailles

1. Cas $d \geq 1$ et $3\,d \leq n$: *répartir d diminutions sur un tour de n mailles*

▪ 1.1.1 Exemple numérique: répartir 15 dimin. sur un tour de 80 mailles

Il reste 80-2*15=50 mailles hors diminutions qu'on répartit en 15 groupes de mailles.
50 divisé par 15 donne 3, reste 5. On peut donc former 5 groupes de 3+1=4 m. et 15-5=10 groupes de 3 m.
Chacun des 15 groupes de mailles sera suivi d'une diminution.
Pour faire le lien avec l'algorithme ci-dessous,

n1=50=nombre de mailles à répartir en groupes;
k1=15=nombre total de groupes à former;
r1=5=(nombre de groupes de p1=4 mailles);
s1=10=(nombre de groupes de q1=3 mailles).

▪ 1.1.2 Algorithme pour répartir *d* dimin. sur un tour de *n* mailles

Soit d diminutions à répartir sur n mailles. Le nouveau nombre de mailles hors diminutions est $(n-2\,d)$ avec lesquelles on va former d groupes de mailles. Chaque groupe de mailles sera suivi d'une diminution. Afin que chaque groupe contienne au moins une maille, nous exigeons que $n-2\,d \geq d \geq 1$, c'est-à-dire $d \geq 1$ et $n \geq 3\,d$.

```
n1 = n - 2 d;
k1 = d;
q1 = Quotient[n1, k1];
r1 = Mod[n1, k1];
p1 = q1 + 1;
s1 = k1 - r1;
```

Relations : $1 \leq k1 \leq n1$, $q1 \geq 1$, $p1 \geq 2$, $r1 + s1 = k1$, $r1 * p1 + s1 * q1 = n1$

▪ 1.1.3 Traitement et affichage des cas particuliers

La procédure groupe1(nmailles) écrit le texte "nmailles m., 1 dimin.".
La procédure assemble(nfois, t) écrit le texte "nfois x(t)".
La procédure alterne(nfois, t1, t2) écrit le texte "répéter nfois fois [t1, t2]".

```
groupe1[nmailles_] := ToString[nmailles] <> " m., 1 dimin.";
assemble[nfois_, t_] := Which[
   nfois == 1, t,
   nfois >= 2, ToString[nfois] <> "x(" <> t <> ")"];
alterne[nfois_, t1_, t2_] := Which[
   nfois == 1, Print[t1, ", ", t2],
   nfois >= 2, Print["répéter ", ToString[nfois], " fois [ ", t1, ", ", t2, " ]"]];
```

Si r1=0, alors n1=s1*q1;
si s1=0, alors n1=r1*p1;
si r1=s1, alors n1=r1*(p1+q1);

```
Which[r1 == 0, Print[assemble[s1, groupe1[q1]]]; Exit[],
    s1 == 0, Print[assemble[r1, groupe1[p1]]]; Exit[],
    r1 == s1, alterne[r1, groupe1[p1], groupe1[q1]]; Exit[]]
```

1.2.1 Exemple numérique: intercaler 5 (groupes de 4 m.) parmi 10 (groupes de 3 m.)

Avec les 10 (groupes de 3 m.), former 5 assemblages de (groupes de 3 m.).
10 divisé par 5 donne 2, reste 0. On peut donc former 0 assemblage de 2+1=3 (groupes de 3 m.) et 5-0=5 assemblages de 2 (groupes de 3 m.).
Chacun des 5 assemblages sera suivi d'un (groupe de 4 m.).
Pour faire le lien avec l'algorithme ci-dessous,

n2=10=nombre de (groupes de 3 m.);
k2=5=nombre total d'assemblages à former;
r2=0=(nombre d'assemblages de p2=3 groupes);
s2=5=(nombre d'assemblages de q2=2 groupes).

- ### 1.2.2 Algorithme pour intercaler *r1* groupes parmi *s1* groupes

Hypothèse : $1 \leq r1 < s1$, sinon permuter (r1, p1) et (s1, q1) :

```
If[r1 > s1, e = r1; r1 = s1; s1 = e; e = p1; p1 = q1; q1 = e]; n2 = s1;
k2 = r1;
q2 = Quotient[s1, k2];
r2 = Mod[s1, k2];
p2 = q2 + 1;
s2 = k2 - r2;
```

Relations : $1 \leq k2 < n2$, $q2 \geq 1$, $p2 \geq 2$, $r2 + s2 = k2$, $r2 * p2 + s2 * q2 = n2$

- ### 1.3.1 Exemple numérique: interpréter et afficher

D'après 1.2.1, on a

répéter 5 fois [2 groupes de 3 m.]

Chaque assemblage est suivi d'un groupe de 4 m.

répéter 5 fois [2 groupes de 3 m., 1 groupe de 4 m.]

Chaque groupe est suivi d'une diminution

répéter 5 fois [2 x (3 m., 1 dimin.), 1 x (4 m., 1 dimin.)]

- ### 1.3.2 Algorithme de l'affichage

```
alterne[r2, assemble[p2, groupe1[q1]], groupe1[p1]];
alterne[s2, assemble[q2, groupe1[q1]], groupe1[p1]]
```

2. Cas $d \geq 1$ et $2d + 1 \leq n \leq 3d$

- ### 2.1.1 Exemple numérique: répartir 15 diminutions sur un tour de 41 mailles

Il reste 41-2*15=11 mailles hors diminutions. Les 15 diminutions sont réparties en 11 groupes.
15 divisé par 11 donne 1, reste 4. On peut donc former 4 groupes de 1+1=2 dimin. et 11-4=7 groupes de 1 dimin.
Chacun des 11 groupes de diminutions est précédé d'une maille.
Pour faire le lien avec les formules qui suivent:

n1=15=nombre de diminutions à répartir en groupes;
k1=11=nombre total de groupes à former;
r1=4=nombre de groupes de 2 dimin.;
s1=7=nombre de groupes de 1 dimin.

2.1.2 Algorithme pour répartir *a* diminutions sur un tour de *n* mailles

Soit d diminutions à répartir sur n mailles. Il reste $(n-2d)$ mailles hors diminutions. Les d diminutions sont réparties en $(n-2d)$ groupes. Chacun des $(n-2d)$ groupes de diminutions est précédé d'une maille. Afin que chaque groupe contienne au moins une diminution, nous exigeons que $d \geq n-2d \geq 1$, c'est-à-dire $d \geq 1$ et $2d+1 \leq n \leq 3d$.

```
n1 = d;
k1 = n - 2 d;
q1 = Quotient[n1, k1];
r1 = Mod[n1, k1];
p1 = q1 + 1;
s1 = k1 - r1;
```

Relations : $1 \leq k1 \leq n1$, $q1 \geq 1$, $p1 \geq 2$, $r1 + s1 = k1$, $r1 * p1 + s1 * q1 = n1$

2.1.3 Traitement et affichage des cas particuliers

La procédure groupe2(ndimin) écrit le texte "1 m., ndimin dimin.".

```
groupe2[ndimin_] := "1 m., " <> ToString[ndimin] <> " dimin."
```

Si r1=0, alors n1=s1*q1;
si s1=0, alors n1=r1*p1;
si r1=s1, alors n1=r1*(p1+q1);

```
Which[r1 == 0, Print[assemble[s1, groupe2[q1]]]; Exit[],
    s1 == 0, Print[assemble[r1, groupe2[ p1]]]; Exit[],
    r1 == s1, alterne[r1, groupe2[p1], groupe2[q1]]; Exit[]]
```

2.2.1 Exemple numérique: intercaler 4 (groupes de 2 dimin.) parmi 7 (groupes de 1 dimin.)

Avec les 7 (groupes de 1 dimin.), former 4 assemblages de (groupes de 1 dimin.).
7 divisé par 4 donne 1, reste 3. On peut donc former 3 assemblages de 1+1=2 (groupes de 1 dimin.) et 4-3=1 assemblage de 1 (groupe de 1 dimin.).
Chacun des 4 assemblages est suivi d'un groupe de 2 dimin.
Pour faire le lien avec l'algorithme ci-dessous,

n2=7=nombre de (groupes de 1 dimin.);
k2=4=nombre total d'assemblages à former;
r2=3=(nombre d'assemblages de p2=2 groupes);
s2=1=(nombre d'assemblages de q2=1 groupe).

2.2.2 Algorithme pour intercaler *r1* groupes parmi *s1* groupes

Hypothèse : $1 \leq r1 < s1$, sinon permuter $(r1, p1)$ et $(s1, q1)$:

```
If[r1 > s1, e = r1; r1 = s1; s1 = e; e = p1; p1 = q1; q1 = e];
n2 = s1;
k2 = r1;
q2 = Quotient[n2, k2];
r2 = Mod[n2, k2];
p2 = q2 + 1;
s2 = k2 - r2;
```

Relations : $1 \leq k2 < n2$, $q2 \geq 1$, $p2 \geq 2$, $r2 + s2 = k2$, $r2 * p2 + s2 * q2 = n2$

2.3.1 Exemple numérique: interpréter et afficher

D'après 1.2.1, on a
    répéter 3 fois [2 groupes de 1 dimin.]
    répéter 1 fois [1 groupe de 1 dimin.]
Chaque assemblage est suivi d'un groupe de 2 dimin.
    répéter 3 fois [2 groupes de 1 dimin., 1 groupe de 2 dimin.]
    1 groupe de 1 dimin., 1 groupe de 2 dimin.
Chaque groupe de dimin. est précédé d'une maille
    répéter 3 fois [2 x (1 m., 1 dimin.), 1 x (1 m., 2 dimin.)]
    1 x (1 m., 1 dimin.), 1 x (1 m., 2 dimin.)

2.3.2 Algorithme de l'affichage

```
alterne[r2, assemble[p2, groupe2[q1]], groupe2[p1]];
alterne[s2, assemble[q2, groupe2[q1]], groupe2[p1]]
```

3. Liens hypertextes

3.1 Calculateur en ligne

http://www.deleze.name/antoinette/TravauxManuels/Tricot/index.html

3.2 Mathématiques pour le tricot

http://www.deleze.name/marcel/culture/tricot/index.html