Tricot à plat:
répartition de *d* diminutions sur un rang de *n* mailles

1. Cas $0 \leq 3\,d + 1 \leq n$: *répartir d diminutions sur un rang de n mailles*

1.1.1 Exemple numérique: répartir 28 dimin. sur un rang de 120 mailles

Il reste 120-2*28=64 mailles hors diminutions qu'on répartit en 28+1=29 groupes de mailles.
64 divisé par 29 donne 2, reste 6. On peut donc former 6 groupes de 2+1=3 m. et 29-6=23 groupes de 2 m.
Les 28 diminutions sont à intercaler entre ces 29 groupes.
Pour faire le lien avec l'algorithme ci-dessous,

n1=64=nombre de mailles à répartir en groupes;
k1=29=nombre total de groupes à former;
r1=6=(nombre de groupes de p1=3 mailles);
s1=23=(nombre de groupes de q1=2 mailles).

1.1.2 Algorithme pour répartir *d* dimin. sur un rang 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 $(d + 1)$ groupes de mailles. Entre deux groupes de mailles, on placera une diminution. Afin que chaque groupe contienne au moins une maille, nous exigeons que $n - 2\,d \geq d + 1 \geq 1$, c'est-à-dire $n \geq 3\,d + 1$.

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

```
Relations : 1 ≤ k1 ≤ n1 , q1 ≥ 1, p1 ≥ 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."; exception: le groupe terminal est "nmailles m.";
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.";
groupe1term[nmailles_] := ToString[nmailles] <> " m.";
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, alterne[1, assemble[s1 - 1, groupe1[q1]], groupe1term[q1]]; Exit[],
    s1 == 0, alterne[1, assemble[r1 - 1, groupe1[p1]], groupe1term[p1]]; Exit[],
```

```
    r1 == s1, alterne[r1 - 1, groupe1[p1], groupe1[q1]]; alterne[1, groupe1[p1],
groupe1term[q1]]; Exit[]]
```

1.2.1 Exemple numérique: intercaler 6 (groupes de 3 m.) parmi 23 (groupes de 2 m.)

Avec les 23 (groupes de 2 m.), former 6+1=7 assemblages de (groupes de 2 m.).
23 divisé par 7 donne 3, reste 2. On peut donc former 2 assemblage de 3+1=4 (groupes de 2 m.) et 7-2=5 assemblages de 3 (groupes de 2 m.).
Les 6 (groupes de 3 m.) sont à intercaler entre les 7 assemblages.
Pour faire le lien avec l'algorithme ci-dessous,

n2=23=nombre de (groupes de 2 m.);
k2=7=nombre total d'assemblages à former;
r2=2=(nombre d'assemblages de p2=4 groupes);
s2=5=(nombre d'assemblages de q2=3 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 + 1;
q2 = Quotient[s1, k2];
r2 = Mod[s1, k2];
p2 = q2 + 1;
s2 = k2 - r2;
```

Relations : $r2 + s2 = r1 + 1$, $r2 * p2 + s2 * q2 = s1$

1.3.1 Exemple numérique: interpréter et afficher

D'après 1.2.1, on a

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

Entre chaque assemblage, intercalons un groupe de 3 m.:

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

Entre chaque groupe, intercalons une augmentation:

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

1.3.2 Algorithme de l'affichage

```
If[s2 >= 1,
 alterne[r2, assemble[p2, groupe1[q1]], groupe1[p1]];
 alterne[s2 - 1, assemble[q2, groupe1[q1]], groupe1[p1]];
 If[q2 > 1, alterne[1, assemble[q2 - 1, groupe1[q1]], groupe1term[q1]],
  Print[groupe1term[q1]]]
 ,
 alterne[r2 - 1, assemble[p2, groupe1[q1]], groupe1[p1]];
 alterne[1, assemble[p2 - 1, groupe1[q1]], groupe1term[q1]]]
```

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

2.1.1 Exemple numérique: répartir 16 diminutions sur un rang de 43 mailles

Il reste 43-2*32=11 mailles hors diminutions. Les 16 diminutions sont réparties en 11-1=10 groupes de mailles.
16 divisé par 10 donne 1, reste 6. On peut donc former 6 groupes de 1+1=2 dimin. et 10-6=4 groupes de 1 dimin.
Les 11 mailles hors diminutions encadrent les 10 groupes de diminutions.
Pour faire le lien avec les formules qui suivent:

n1=16=nombre de diminutions à répartir en groupes;
k1=10=nombre total de groupes à former;
r1=6=nombre de groupes de 2 dimin.;
s1=4=nombre de groupes de 1 dimin.

2.1.2 Algorithme pour répartir *a* augmentations sur un rang de *n* mailles

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

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

```
Relations : 1 ≤ k1 ≤ n1 , q1 ≥ 1, p1 ≥ 2, r1 + s1 = k1, r1 * p1 + s1 * q1 = n1
```

2.1.3 Traitement et affichage des cas particuliers

La procédure groupe2(naugm) écrit le texte "1 m., ndimin dimin."; exception: le groupe terminal est "1 m., ndimin dimin., 1 m.";

```
groupe2[ndimin_] := "1 m., " <> ToString[ndimin] <> " dimin.";
groupe2term[ndimin_] := groupe2[ndimin] <> ", 1 m.";
```

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, alterne[1, assemble[s1 - 1, groupe2[q1]], groupe2term[q1]]; Exit[],
    s1 == 0, alterne[1, assemble[r1 - 1, groupe2[p1]], groupe2term[p1]]; Exit[],
    r1 == s1, alterne[r1 - 1, groupe2[p1], groupe2[q1]]; alterne[1, groupe2[p1],
groupe2term[q1]]; Exit[]]
```

2.2.1 Exemple numérique: intercaler 4 (groupes de 1 dimin.) parmi 6 (groupes de 2 dimin.)

Avec les 6 (groupes de 2 dimin.), former 4+1=5 assemblages de (groupes de 2 dimin.).
6 divisé par 5 donne 1, reste 1. On peut donc former 1 assemblage de 1+1=2 (groupes de 2 dimin.) et 5-1=4 assemblages de 1 (groupe de 2 dimin.).
Les 4 (groupes de 1 dimin.) seront intercalés entre les 5 assemblages.
Pour faire le lien avec l'algorithme ci-dessous,

n2=6=nombre de (groupes de 2 dimin.);
k2=5=nombre total d'assemblages à former;
r2=1=(nombre d'assemblages de p2=2 groupes);
s2=4=(nombre d'assemblages de q2=1 groupe).

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

Hypothèse : $1 \le r1 < s1$, 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 + 1;
q2 = Quotient[n2, k2];
r2 = Mod[n2, k2];
p2 = q2 + 1;
s2 = k2 - r2;
```

Relations : $2 \le k2 \le n2$, $q2 \ge 1$, $p2 \ge 2$, $r2 + s2 = k2$, $r2 * p2 + s2 * q2 = n2$

2.3.1 Exemple numérique: interpréter et afficher

D'après 1.2.1, on a

répéter 1 fois [2 groupes de 2 dimin.]
répéter 4 fois [1 groupe de 2 dimin.]

Entre chaque assemblage, intercalons un groupe de 1 dimin.:

2 groupes de 2 dimin., 1 groupe de 1 dimin.
répéter 3 fois [1 groupe de 2 dimin., 1 groupe de 1 dimin.]
1 groupe de 2 dimin.

Encadrons chaque groupe par des mailles

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

2.3.2 Algorithme de l'affichage

```
If[s2 >= 1,
 alterne[r2, assemble[p2, groupe2[q1]], groupe2[p1]];
 alterne[s2 - 1, assemble[q2, groupe2[q1]], groupe2[p1]];
 If[q2 > 1, alterne[1, assemble[q2 - 1, groupe2[q1]],
   groupe2term[q1]],
  Print[groupe2term[q1]]]
 ,
 alterne[r2 - 1, assemble[p2, groupe2[q1]], groupe2[p1]];
 alterne[1, assemble[p2 - 1, groupe2[q1], groupe2term[q1]]]]
```

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