

# Tricot à plat: répartition de $a$ augmentations sur un rang de $n$ mailles

## 1. Cas $0 \leq a < n$ : répartir $a$ augmentations sur un rang de $n$ mailles

### ■ 1.1.1 Exemple numérique: répartir 36 augm. sur un rang de 116 mailles

Avec les 116 m., former  $36+1=37$  groupes de mailles.

116 divisé par 37 donne 3, reste 5. On peut donc former 5 groupes de  $3+1=4$  mailles et  $37-5=32$  groupes de 3 mailles.

Les 36 augmentations sont à intercaler entre ces 37 groupes.

Pour faire le lien avec l'algorithme ci-dessous,

$n1=116$ =nombre de mailles à répartir en groupes;

$k1=37$ =nombre total de groupes à former;

$r1=5$ =(nombre de groupes de  $p1=4$  mailles);

$s1=32$ =(nombre de groupes de  $q1=3$  mailles).

### ■ 1.1.2 Algorithme pour répartir $a$ augm. sur un rang de $n$ mailles

Soit  $a$  augmentations à répartir sur  $n$  mailles. Avec ces  $n$  mailles, on va former  $(a + 1)$  groupes de mailles entre lesquels on placera  $a$  augmentation. Afin que chaque groupe contienne au moins une maille, nous exigeons que  $n \geq a + 1 \geq 1$ , c'est-à-dire  $0 \leq a < n$ .

---

```
n1 = n;  
k1 = a + 1;  
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 " $n$  mailles m., 1 augm."; exception: le groupe terminal est " $n$  mailles m.";

La procédure `assemble(nfois, t)` écrit le texte " $n$  fois  $x(t)$ ".

La procédure `alterne(nfois, t1, t2)` écrit le texte "répéter  $n$  fois [ $t1, t2$ ]".

---

```
groupe1[nmailles_] := ToString[nmailles] <> " m., 1 augm.";  
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 5 (groupes de 4 m.) parmi 32 (groupes de 3 m.)

Avec les 32 (groupes de 3 m.), former  $5+1=6$  assemblages de (groupes de 3 m.).

32 divisé par 6 donne 5, reste 2. On peut donc former 2 assemblages de  $5+1=6$  (groupes de 3 m.) et  $6-2=4$  assemblages de 5 (groupes de 3 m.).

Les 5 groupes de 4 m. sont à intercaler entre les 6 assemblages.

Pour faire le lien avec l'algorithme ci-dessous,

```

n2=32=nombre de (groupes de 3 m.);
k2=6=nombre total d'assemblages à former;
r2=2=(nombre d'assemblages de p2=6 groupes);
s2=4=(nombre d'assemblages de q2=5 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[n2, k2];
r2 = Mod[n2, k2];
p2 = q2 + 1;
s2 = k2 - r2;

```

---

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

### ■ 1.3.1 Exemple numérique: interpréter et afficher

D'après 1.2.1, on a

répéter 2 fois [6 groupes de 3 m.]

répéter 4 fois [5 groupes de 3 m.]

Entre chaque assemblage, intercalons un groupe de 4 m.:

répéter 2 fois [6 groupes de 3 m., 1 groupe de 4 m.]

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

5 groupes de 3 m.

Entre chaque groupe, intercalons une augmentation:

répéter 2 fois [6 x (3 m., 1 augm.), 1 x (4 m., 1 augm.)]

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

4 x (3 m., 1 augm.)

3 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 $2 \leq n \leq a+1$ : répartir $a$ augmentations sur un rang de $n$ mailles

### ■ 2.1.1 Exemple numérique: répartir 12 augmentations sur un rang de 8 mailles

Avec les 12 augm., former  $8-1=7$  groupes d'augmentations.

12 divisé par 7 donne 1, reste 5. On peut donc former 5 groupes de  $1+1=2$  augm. et  $7-5=2$  groupes de 1 augm.

Les 8 mailles encadrent les 7 groupes d'augmentations.

Pour faire le lien avec les formules qui suivent:

$n1=116$ =nombre d'augmentations à répartir en groupes;

$k1$ =nombre total de groupes à former;

$r1$ =nombre de groupes d'effectifs  $p1$ ;

$s1$ =nombre de groupes d'effectifs  $q1$ .

### ■ 2.1.2 Algorithme pour répartir $a$ augmentations sur un rang de $n$ mailles

Avec les  $a$  augmentations, on va former  $(n-1)$  groupes d'augmentations. Entre deux mailles, on placera un groupe d'augmentations. Afin que chaque groupe contienne au moins une augmentation, on exige que  $a \geq n-1 \geq 1$ , c'est-à-dire  $2 \leq n \leq a+1$ .

---

```
n1 = a;
k1 = n - 1;
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(naugm) écrit le texte "1 m., naugm augm."; exception: le groupe terminal est "1 m., naugm augm., 1 m.".

---

```
groupe2[naugm_] := "1 m., " <> ToString[naugm] <> " augm.";
groupe2term[naugm_] := groupe2[naugm] <> ", 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 2 (groupes de 1 augm.) parmi 5 (groupes de 2 augm.)

Avec les 5 (groupes de 2 augm.), former  $2+1=3$  assemblages de (groupes de 2 augm.).

5 divisé par 3 donne 1, reste 2. On peut donc former 2 assemblages de  $1+1=2$  (groupes de 2 augm.) et  $3-1=1$  assemblage de 1 (groupe de 2 augm.).

Les 2 groupes de 1 augm. seront intercalés entre les 3 assemblages.

Pour faire le lien avec l'algorithme ci-dessous,

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

---

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

### ■ 2.3.1 Exemple numérique: interpréter et afficher

D'après 2.2.1, on a

répéter 2 fois [2 groupes de 2 augm.]  
 1 fois [1 groupe de 2 augm.]

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

répéter 2 fois [2 groupes de 2 augm., 1 groupe de 1 augm.]  
 1 groupe de 2 augm.

Encadrons chaque groupe d'augm. avec des mailles

répéter 2 fois [2 x (1 m., 2 augm.), 1 x (1 m., 1 augm.)]  
 1 x (1 m., 2 augm.)  
 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>