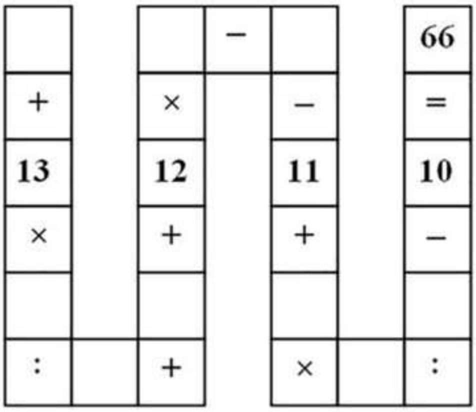
# Un problème de calcul au Vietnam

Voici le casse-tête de mathématiques posé par le professeur Tran Phuong aux élèves de sa classe (niveau CE2), à Bao Loc au Vietnam, problème qui a fait le tour du monde en mai 2015.




Il s'agit d'une addition dont le résultat est 66, avec 9 nombres manquants (indiqués par un carré noir ci-dessous) qu'il s'agit de trouver, sachant que ces 9 nombres ne sont autres que les chiffres 1, 2, 3, ..., 8, 9, à n'utiliser qu'une fois chacun.

$$\blacksquare + \frac{13\times\blacksquare}{\blacksquare} + \blacksquare + 12\times\blacksquare - \blacksquare - 11 + \frac{\blacksquare\times\blacksquare}{\blacksquare} - 10 = 66$$

ou encore :

$$\blacksquare + \frac{13\times\blacksquare}{\blacksquare} + \blacksquare + 12\times\blacksquare - \blacksquare + \frac{\blacksquare\times\blacksquare}{\blacksquare} = 87$$

En notant $a_i$ (avec $i$ de 1 à 9) les nombres manquants, cela s'écrit :

$$a_1 + \frac{13\times a_2}{a_3} + a_4 + 12\times a_5 - a_6 + \frac{a_7\times a_8}{a_9} = 87$$

Il s'agit de trouver $a_1\ a_2\ a_3\ \dots\ a_9$ qui est une permutation de 1 2 3 ... 9. Pour cela reprenons le programme des permutations.[1] Les permutations de 9 éléments sont au nombre de 362 880. Il suffit d'afficher celles qui obéissent à l'égalité précédente, et aussi telles que les deux divisions présentes tombent justes.[2] On en déduit le programme :

```
#include <stdio.h>
int a[10];

int main()
{int i,pospivot,posremplacant,aux,gauche,droite;
int compteur=0;
  for(i=1;i<=9;i++) a[i]=i;  /* le tableau a[] possède 10 cases, dont nous n'utilisons pas la case 0 */
  for(;;)
     { if (a[2]%a[3]==0 && (a[7]*a[8])%a[9]==0 && a[3]!=1 && a[9]!=1
           && a[1]+13*a[2]/a[3]+a[4]+12*a[5]-a[6]+(a[7]*a[8])/a[9]==87)
```

[1] Voir sur mon site *audibertpierre.fr*, rubrique Enseignements, Cours d'algorithmique, chapitre 7 : énumérations dans l'ordre alphabétique.

[2] On a aussi supposé qu'aucun des diviseurs $a_3$ et $a_9$ ne soit égal à 1, car sinon une des divisions deviendrait ridicule. Toutefois en acceptant ce 1 en diviseur, on trouverait 20 solutions au lieu de 4.

```
            {compteur++; printf("\n %d* ",compteur); for(i=1; i<=9; i++) printf(" %d ",a[i]); }
        i=9; while(i>1 && a[i]<a[i-1]) i--;
        pospivot=i-1;   if (pospivot==0) break;
        i=9;while(a[i]<a[pospivot]) i--; posremplacant=i;
        aux=a[pospivot];a[pospivot]=a[posremplacant]; a[posremplacant]=aux;
        gauche=pospivot+1; droite=9;
        while (gauche<droite)
           {aux=a[gauche]; a[gauche]=a[droite]; a[droite]=aux; gauche++;droite--;}
      }
  getchar() ;return 0;
     }
```

L’exécution du programme, en quelques centièmes de seconde, donne quatre solutions pour les $a_i$ :

```
1*    5  9  3  6  2  1  7  8  4
2*    5  9  3  6  2  1  8  7  4
3*    6  9  3  5  2  1  7  8  4
4*    6  9  3  5  2  1  8  7  4
```