

(d'après Symbole TS BELIN)

## Énoncé

Un site Web propose des jeux en ligne. Parmi eux, celui-ci :

- Si l'internaute gagne une partie, la probabilité qu'il gagne la partie suivante est  $\frac{2}{5}$ .
- Si l'internaute perd une partie, la probabilité qu'il perde la partie suivante est  $\frac{4}{5}$ .

Pour tout entier naturel non nul  $n$ , on désigne par  $G_n$  l'événement « l'internaute gagne la  $n^{\text{ième}}$  partie » et on note  $p_n$  la probabilité de l'événement  $G_n$ . L'internaute gagne toujours la première partie et donc  $p_1 = 1$ .

### 1. Simulation

On considère l'algorithme suivant :

```
Saisir n.
R[1] = 1
Pour i allant de 2 à n
  Simuler un aléa entre 0 et 1 et le stocker dans X
  Si R[i-1] = 1, alors
    Si X ≤ 2/5, Faire R[i] = 1 Sinon, R[i] = 0
  Si R[i-1] = 0, alors
    Si X ≤ 1/5, Faire R[i] = 1 Sinon, R[i] = 0
Afficher R
Afficher la moyenne de R
```

- Que signifie l'instruction :  $R[1] = 1$
- Que représente la simulation de R ? Que calcule la moyenne de R ?
- Programmez cet algorithme sur la Class Pad. Exécutez-le pour différentes valeurs de  $n$ .
- Tracez la courbe de la moyenne de R en fonction de  $n$ , que constate-t-on ?

### 2. Des démonstrations

- Construisez un arbre pondéré.
- Montrez que pour tout entier naturel non nul  $n$ ,  $p_{n+1} = \frac{1}{5}p_n + \frac{1}{5}$ .
- On pose  $u_n = p_n - \frac{1}{4}$ .
  - Montrez que la suite  $(u_n)$  est une suite géométrique dont vous préciserez la raison et le premier terme.
  - Donnez  $u_n$  puis  $p_n$  en fonction de  $n$ .
  - Déterminez la limite de la suite  $(p_n)$  puis celle de  $(u_n)$ .
  - Ce résultat est-il en accord avec la simulation de cette expérience aléatoire faite au 1) ?

## Éléments de réponse : Utilisation de la Casio Fx-CP400 pour effectuer la simulation de l'expérience aléatoire.

On notera que dans ce programme les résultats obtenus : 0 ou 1, sont stockés dans une liste R de n éléments.

$R[1] = 1$  signifie que, la première partie étant gagnée, le 1<sup>er</sup> élément de la liste R est un « 1 », un « 0 » signifiant que la partie est perdue.

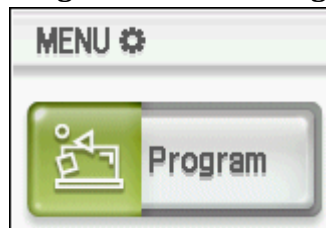
A la i<sup>ème</sup> partie on peut donc avoir :  $R[i] = 1$  ou  $= 0$ .

En effet, c'est la valeur aléatoire obtenue pour X qui détermine si la partie est gagnée ou non.

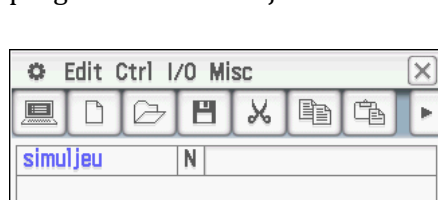
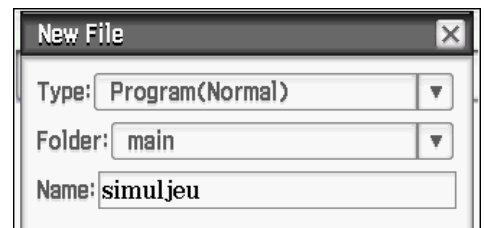
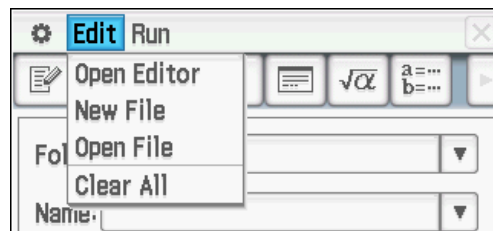
Puis on calcule la moyenne de la série composée des différentes valeurs obtenues: la moyenne de R représente  $p_n$ .

Il semble que les valeurs de  $p_n$  se rapprochent de 0,25 quand n devient « grand ».

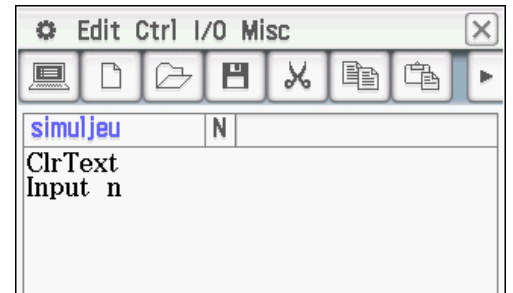
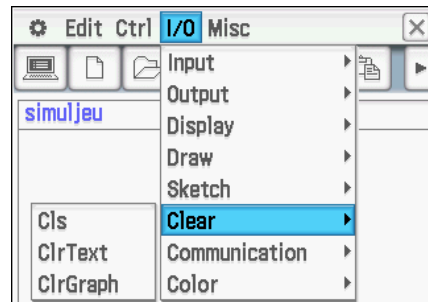
### Programmation de l'algorithme : dans le menu Programme



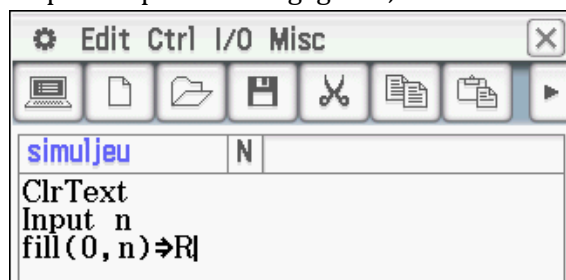
Créer un nouveau programme : « simuljeu »



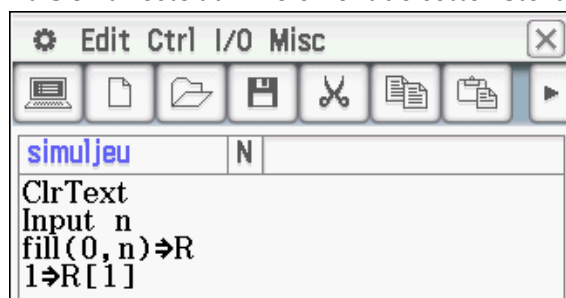
ClrText permet, à chaque nouvelle simulation, d'effacer les valeurs précédentes obtenues.



Pour créer la liste R dans laquelle on va stocker les différentes valeurs 0 ou 1 qui vont donc représenter le nombre de parties perdues ou gagnées, on utilise l'instruction « fill(0,n) » : liste de « longueur n » nommée R :



Puis on affecte au 1<sup>er</sup> élément de cette liste la valeur 1, qui signifie que la 1<sup>ère</sup> partie est gagnée :



Puis on saisit la boucle « For » : en utilisant la « touche » Ctrl

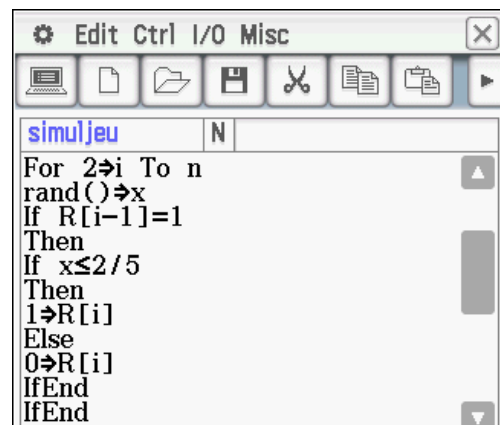
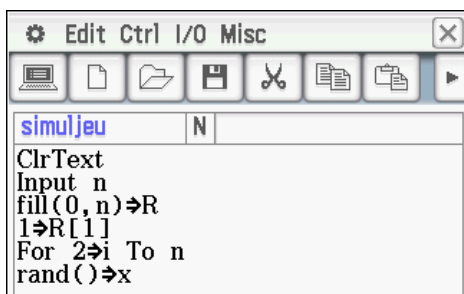
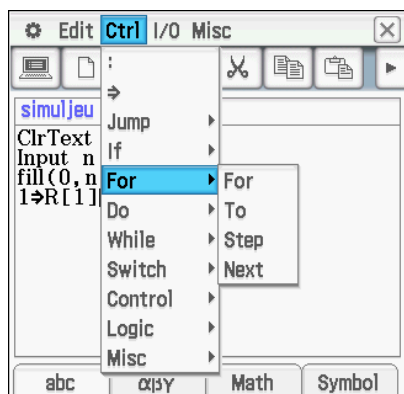
Ne pas oublier à la fin de la « fermer » avec « Next »

L'instruction « rand() » permet d'obtenir un réel aléatoire compris entre 0 et 1.

Dans cette boucle on saisit la 1<sup>ère</sup> condition « If » correspondant à : « si la (i-1)<sup>ème</sup> partie est gagnée alors.... » :

Saisir « If R[i-1]=1 »

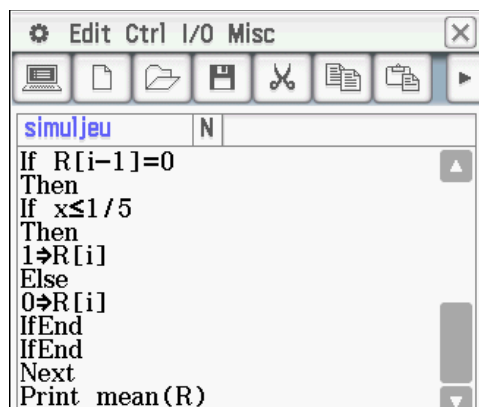
On pense bien à « fermer » les deux « si » : IfEnd, à la ligne, IfEnd.



Pour la 2<sup>ème</sup> condition « If », correspondant à : « si la (i-1)<sup>ème</sup> partie est perdue alors.... » : saisir « If R[i-1]=0 »

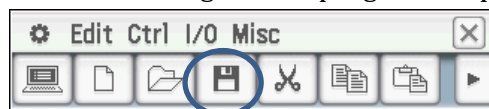
Ne pas hésiter à faire un copié-collé de la 1<sup>ère</sup> en rectifiant ce qui doit l'être dans la 2<sup>ème</sup> !

Sélectionner ce qui doit être copié, aller dans « Edit » puis « Copier » puis « Coller » à la suite et modifier.... On obtient :

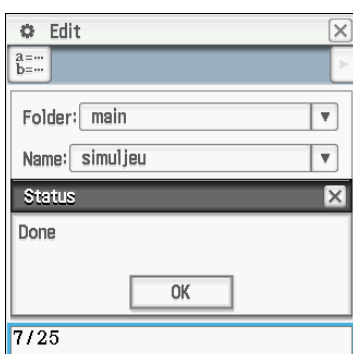
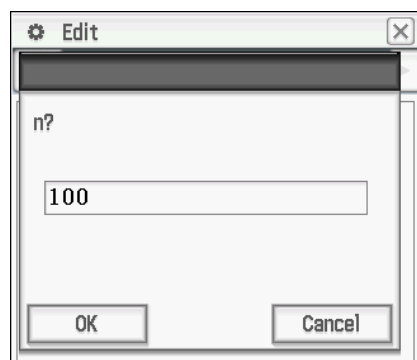


Pour finir, et calculer et afficher la moyenne des valeurs stockées dans la liste R, taper l'instruction « Print mean(R) »

Puis il faut enregistrer le programme, pour cela cliquer sur l'icône « disquette » :



Puis pour lancer le programme et l'exécuter cliquer dans l'ordre sur :



Saisir une valeur de n assez grande pour obtenir la moyenne de R.

On obtient, par exemple, pour n = 100, une moyenne de 0,28, ce qui est assez « proche » de la limite calculée : 0,25

Faire d'autres simulations avec n= 500 etc...