|  |  |
| --- | --- |
| **Stage algorithmique 1****TI graphiques (83 Premium CE & 82 Advanced)** | **Treize**  |

**Le problème :** on se propose de simuler, sur une calculatrice, le jeu suivant :

On lance un dé jusqu’à ce que le total des points obtenus lors des différents lancers soit supérieur ou égal à 13.

Combien faut-il de lancers de dé ?

Évidemment, il faut au moins trois lancers et au plus treize.

**1. Description du programme**

On utilise :

• Trois variables : une nommée S pour le total des points, initialisée à 0, une autre nommée N pour le nombre de lancers, initialisée également à 0 et une nommée A pour stocker provisoirement le résultat de chaque lancer ;

• Une structure répétitive : placer dans A un nombre aléatoire compris entre 1 et 6, afficher A, ajouter A à S, augmenter N de 1, recommencer tant que S est inférieur strictement à 13 ;

• Un affichage final.

**2. Le programme**

|  |  |
| --- | --- |
| Algorithme | Programme : DE13 |
| Mettre 0 dans SMettre 0 dans NTant que S est strictement inférieur à 13 Mettre dans A un nombre aléatoire entre 1 et 6 Afficher A Ajouter A à S Augmenter N de 1Fin du tant queAfficher N  | 0 sto S0 sto N While S<13nbrAléatEnt(1,6) sto ADisp AS + A sto S N + 1 sto NEndDisp "N: ",N |

**3. Prolongement**

Écrire un programme qui donne les résultats de 100 répétitions du programme précédent DE13.

Pour cela, commencer par supprimer les lignes d’affichage du programme DE13 pour en accélérer l’exécution ou le réécrire sans les deux lignes contenant l’instruction DISP.

|  |  |
| --- | --- |
| Algorithme | Programme : DE100 |
| Préparer une liste L1 de 13 élémentsMettre 0 dans chaque élément de L1Mettre 0 dans ITant que I est strictement inférieur à 100 Effectuer le programme DE13 (sans affichage) Ajouter 1 au Nième élément de L1 Augmenter I de 1Fin du tant queAfficher L1 | 13 sto dim(L1)Remplir (0,L1) 0 sto IWhile I < 100**Prgm**  DE13L1(N) + 1 sto L1(N)I + 1 sto IEndL1 |

*Remarques* :

• Écrire **L1** au lieu de **Disp L1** suffit pour obtenir l’affichage en fin de programme.

• La liste **L1** peut être facilement lue dans l’éditeur statistique par **List** 1:Edit.

• Une boucle **For ( I, 1, 100) … End** remplace avantageusement **While … End** puisqu’alors il n’est pas nécessaire d’initialiser ni d’incrémenter la variable I.

**4. Version sans sous-programme utilisant l’instruction For**

|  |  |
| --- | --- |
| Algorithme | Programme : DE100B |
| Préparer une liste L1 de 13 élémentsMettre 0 dans chaque élément de L1Pour I de 1 à 100 Mettre 0 dans S Mettre 0 dans N Tant que S est strictement inférieur à 13 Mettre dans A un nombre aléatoire entre 1 et 6 Ajouter A à S Augmenter N de 1 Fin du tant que Ajouter 1 au Nième élément de L1Fin du ForAfficher L1 | 13 sto dim (L1)Remplir (0, L1) For( I, 1, 100 ) 0 sto S0 sto N While S<13nbrAléatEnt(1,6) sto AS + A sto S N + 1 sto NEndL1(N) + 1 sto L1(N)EndL1 |

*Remarque :*

On obtient directement l’analyse statistique de l’échantillon en modifiant la fin du programme de la manière suivante :

Éditer le programme, modifier la dernière ligne qui contient L1 en : **Pause** L1

Puis ajouter les instructions

**suite**( J, J, 1, 13) sto L2

**Stats** **1-Var** L2, L1

Après affichage de la liste L1, on obtiendra les paramètres statistiques en appuyant sur ***Entrer/enter***.