

```

0001 //-----
0002 //Syracuse - simple
0003 //-----
0004 //Ce programme calcule les termes de la suite de syracuse jusqu'à ce que la suite arrive à 1
0005 //Il affiche alors le temps de vol obtenu
0006
0007 //
0008 //clear // On efface toutes les variables
0009 //u=26623 // On initialise la suite
0010 //u0=u // On garde en mémoire u0
0011 //v=0 //On initialise le temps de vol
0012 //// Boucle de calcul de la suite :
0013 //while u>1
0014 //     v=v+1 //On incrémente le temps de vol
0015 //     //Calcul du terme suivant de la suite
0016 //     if floor(u/2)==u/2 // si u/2 est entier c-a-d si u est pas pair
0017 //         u=u/2
0018 //     else
0019 //         u=3*u+1
0020 //     end
0021 //end
0022 //disp("le temps de vol pour u0 = "+string(u0)+" est de "+string(v)+".")
0023
0024 //-----
0025 //Syracuse - V2
0026 //-----
0027 //Ce programme essaie tous les u0 de n1 à n2
0028 //puis affiche le temps de vol maximal atteint et
0029 //le plus petit u0 pour lequel il est atteint.
0030
0031 //clear //On efface toutes les variables
0032 //vmax=0 //On initialise le temps de vol maxi
0033 //// On définit la valeur min et max de u0 que l'on va tester
0034 //u0min=1
0035 //u0max=10000
0036 //u0VolMax=u0min
0037 ////On rentre dans la boucle où l'on va tester chaque u0
0038 //for u0=u0min:u0max
0039 //     // On affiche la valeur de u0 tous les multiples de 1000 pour savoir où on en est.
0040 //     if floor(u0/1000)==u0/1000
0041 //         disp(u0)
0042 //     end
0043 //     // Début du calcul de la suite de Syracuse
0044 //     // Cette partie est identique au programme Syracuse 1
0045 //     u=u0
0046 //     v=0
0047 //     while u>1
0048 //         v=v+1
0049 //         if floor(u/2)==u/2
0050 //             u=u/2
0051 //         else
0052 //             u=3*u+1
0053 //         end
0054 //     end
0055 //     if v>vmax //si on a obtenu un temps de vol supérieur au temps de vol maxi
0056 //         u0VolMax=u0
0057 //         vmax=v
0058 //     end
0059 //end
0060 //disp("on a obtenu un temps de vol maxi de "+string(vmax) + " pour u0 = "+string(u0VolMax))
0061 //
0062
0063 //-----
0064 //Syracuse - V3
0065 //-----
0066 //Ce programme trace en fonction de u0 le temps de vol observé pour u0 allant de u0min à u0max
0067
0068 clear //On efface toutes les variables
0069 clf//On efface la fenêtre graphique
0070 // On définit la valeur min et max de u0 que l'on va tester
0071 u0min=1
0072 u0max=10000
0073 V=-1*ones(1:u0max-u0min+1) //On initialise la matrice des temps de vol à -1

```

```

0074 //On rentrer dans la boucle où l'on va tester chaque u0
0075 c=0 //On crée un compteur qui permettra de savoir à quel itération on en est.
0076 for u0=u0min:u0max
0077     c=c+1
0078     // On affiche la valeur de u0 tous les multiples de 1000 pour savoir où on en est.
0079     if floor(u0/1000)==u0/1000
0080         disp(u0)
0081     end
0082     // Début du calcul de la suite de Syracuse
0083     // Cette partie est identique au programme Syracuse 1
0084     u=u0
0085     v=0
0086     while u>1
0087         v=v+1
0088         if floor(u/2)==u/2
0089             u=u/2
0090         else
0091             u=3*u+1
0092         end
0093     end
0094     V(c)=v //On enregistre le temps de vol obtenu dans la matrice des temps de vol
0095 end
0096 plot(u0min:u0max,V) //On trace les temps de vol obtenus en fonction de u0 testés.
0097
0098 //disp(uvmax,vmax)
0099
0100 //n=9
0101 //r=2
0102 //b=1
0103 //N=b+r
0104 //nrt=0
0105 //
0106 //for k=1:n
0107 //     bt=grand(1,1,'uin',1,N)
0108 //     if bt<=r
0109 //         r=r+1
0110 //         nrt=nrt+1
0111 //     else
0112 //         b=b+1
0113 //     end
0114 //     N=N+1
0115 //end
0116 //disp(nrt)
0117
0118 //clear
0119 //n=90
0120 //nrtT=zeros(1,1000)
0121 //for j=1:1000
0122 //     r=2
0123 //     b=1
0124 //     N=b+r
0125 //     nrt=0
0126 //
0127 //     for k=1:n
0128 //         bt=grand(1,1,'uin',1,N)
0129 //         if bt<=r
0130 //             r=r+1
0131 //             nrt=nrt+1
0132 //         else
0133 //             b=b+1
0134 //         end
0135 //         N=N+1
0136 //     end
0137 //     nrtT(j)=nrt
0138 //end
0139 //disp(sum(nrtT)/1000)
0140
0141 //n=10
0142 //tirage = grand(1,1,"uin",1,n)
0143 //k=1
0144 //while tirage >2
0145 //     k=k+1
0146 //     tirage = grand(1,1,"uin",1,n)

```

```
0147 //end
0148 //X=k
0149 //disp(X)
0150 //n=20
0151 //S=0
0152 //N=10000
0153 //for i=1:N
0154 //     nr=n
0155 //     tirage = grand(1,1,"uin",1,nr)
0156 //     k=1
0157 //     while tirage >2
0158 //         k=k+1
0159 //         nr=nr-1
0160 //         tirage = grand(1,1,"uin",1,nr)
0161 //     end
0162 //     S=S+k
0163 //end
0164 //disp(S/N)
```