Nom : ……
Prénom : ……

Partie II

Exercice 2 (5 pts)

La suite de Syracuse est définie par :

$$u_{n+1} = \begin{cases} {}^{u_n}/_2 & si\ u_n\ est\ pair, \\ 3u_n + 1 & sinon \end{cases}$$

La conjecture de Syracuse affirme que, quel que soit le terme initial u_0 (non nul) de la suite, celle-ci finit par valoir 1 (puis boucler sur 4, 2, 1).
Etant donnée une valeur initiale u_0 de la suite, on appelle *temps de vol*, le plus petit indice k pour lequel $u_k = 1$ pour la première fois, et *altitude maximale* la valeur maximale prise par la suite durant ce temps de vol.
Par exemple, pour $u_0 = 14$, on construit la suite des nombres :

14, 7, 22, 11, 34, 17, **52**, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, **1**, 4, 2,1,…

Le temps de vol dans cet exemple est de 17 et l'altitude maximale est de 52.

1. Ecrire un programme qui exécute une boucle infinie dans laquelle on fixe le terme initial u_0 de manière aléatoire, puis on calcule les termes de la suite de Syracuse jusqu'à ce qu'elle vaille 1. Le programme boucle indéfiniment sans rien afficher.
2. En utilisant l'interface POSIX de gestion des signaux :
   - Faire en sorte qu'à la réception du signal SIGTSTP (Ctrl-Z), le programme affiche :
     - la valeur initiale courante,
     - la valeur et l'indice du dernier terme calculé.
   - Faire en sorte qu'à la réception du signal SIGINT (Ctrl-C), le programme affiche :
     - la plus grande valeur initiale testée ;
     - le plus grand temps de vol atteint et la valeur initiale pour laquelle ce dernier est atteint ;
     - la plus grande altitude maximale atteinte et la valeur initiale pour laquelle cette dernière est atteinte.

     Si le programme reçoit dans les trois secondes un autre signal SIGINT et si le signal SIGTSTP a été reçu au moins 10 fois, alors le programme doit se terminer.

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

/*****************************************/
/***          Variables globales       ***/   (0.25 pt)
/*****************************************/

unsigned long int u0, uk, k;
unsigned long int tvol_max=0, u0_tvol_max;
unsigned long int alt_max=0, u0_alt_max;
struct sigaction act_sigint, act_sigstop, act_alarm;
unsigned int nb_sigstop = 0;
int in_sigint = 0;
```

Nom : ……
Prénom : ……

```
/*****************************************/
/***                Prototypes des fonctions                ***/   (0.25 pt)
/*****************************************/
void handler_sigstop(int signum);
void handler_sigint(int signum);
void handler_alarm(int signum);


/*****************************************/
/***                Programme principal                ***/   (1.5 pts)
/*****************************************/
main(){
   act_sigstop.sa_handler = handler_sigstop;
   sigaction(SIGTSTP, &act_sigstop, NULL);
   act_sigint.sa_handler = handler_sigint;
   sigaction(SIGINT, &act_sigint, NULL);
   act_alarm.sa_handler = handler_alarm;
   sigaction(SIGALRM, &act_alarm, NULL);

   while(1){
      u0 = random()%2000+1;
      uk = u0; k = 0;
      while(uk != 1){
         if(uk > alt_max){
            alt_max = uk;
            u0_alt_max = u0;
         }
         if(uk%2) uk = 3*uk+1;
         else uk = uk/2;
         k++;
      }
      if(k > tvol_max){
         tvol_max = k;
         u0_tvol_max = u0;
      }
   }
}

/*****************************************/
/***                Définition des fonctions                ***/   (3 pts)
/*****************************************/
void handler_sigstop(int signum){
   printf("\n\n u(0) = %d \t u(%d) = %d\n", u0,k,uk);
   nb_sigstop++;
}
void handler_sigint(int signum){
   if(!in_sigint){
      printf("\n\nDernier u(0) = %d \n", u0);
      printf("Temps de vol max = %d \t u(0) = %d\n", tvol_max,  u0_tvol_max);
      printf("Altitude max = %d \t u(0) = %d\n", alt_max, u0_alt_max);
      in_sigint = 1;
      alarm(2);
   }
   else if(nb_sigstop >= 5)  exit(0);
}
void handler_alarm(int signum){
   in_sigint = 0;
}
```