Beispiellösung: Buffet-Lotterie

Hinweis:
Der Aufgabentext wird hier nur der Vollständigkeit halber abgedruckt. Die Dokumentation zu einer Aufgabenbearbeitung muss und soll den Aufgabentext nicht enthalten.

Bei der Endrunde des Bundeswettbewerbs Informatik haben die Teilnehmer es satt, Warteschlangenfutter vor dem großen Buffet im engen Korridor und Opfer der Last-Come-Longest-Hungry-Mentalität zu sein. Stattdessen soll ganz elegant und zivilisiert ausgelost werden, wer als Nächster das Buffet aufsuchen darf.

Die 28 Teilnehmer stellen sich dazu in einem großen Kreis auf und sagen den Satz

In-for-ma-tik kann uns wei-sen,
wer als Nächs-ter kommt zum Spei-sen

wiederholt laut auf. Wie beim Ene-Mene-Muh spricht jeder Teilnehmer nur eine Silbe, dann ist sein rechter Nachbar an der Reihe. Und wer die letzte Silbe des Satzes sagt, ist der Glückliche, der den Kreis verlassen und als Nächster seinen Hunger stillen darf.

Eine Teilnehmerin hat aber Geburtstag. Sie spricht natürlich die allererste Silbe, und als besondere Gunst darf sie, wann immer sie an der Reihe ist, statt einer auch zwei Silben sprechen – wenn sie das denn will. Da ihr Magen knurrt, möchte sie ihren Vorteil dazu nutzen, so schnell wie möglich zum Buffet zu kommen.

Aufgabe

Schreibe ein Programm, das für eine gegebene Anzahl von Teilnehmern berechnet, wann das Geburtstagskind zwei Silben sprechen soll, um sich den bestmöglichen Platz in der Buffetreihenfolge zu verschaffen.

Dokumentiere die Wirkungsweise deines Programms für verschiedene Teilnehmerzahlen, unter anderem für die oben genannten 28 Teilnehmer.

Lösungsidee

Simulation

Das Programm soll die Buffet-Lotterie „nachspielen", also simulieren. Dazu muss genauer beschrieben werden, was bei der Buffet-Lotterie passiert – zunächst einmal ohne den Vorteil für das Geburtstagskind, das zur Vereinfachung im Folgenden Gina heißt:

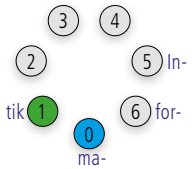
> Die Anzahl der Teilnehmer nennen wir T. In der Aufgabe ist von 28 Teilnehmern die Rede ($T = 28$), aber die Lösung soll auch für andere Werte funktionieren.

> Die Teilnehmer, die noch nicht zum Buffet gegangen sind, stehen im Kreis. Zu Beginn sind das alle T Teilnehmer, aber nach jedem Sprechen des Satzes geht ein Teilnehmer zum Buffet, und es bleiben R Teilnehmer (R wie Rest) im Kreis übrig. Diese R Teilnehmer haben die Nummern 0 bis R–1. Verlässt ein Teilnehmer den Kreis, rücken die Teilnehmer mit höherer Nummer auf und bekommen nun neue, um 1 niedrigere Nummern. Für die Lösung ist es am einfachsten, wenn Gina die Nummer 0 hat. Sie kann dann nämlich garantiert ihre Nummer behalten, bis sie an der Reihe ist.

> Die Anzahl der Silben, die zu sprechen sind, nennen wir S. Der Satz in der Aufgabe hat 16 Silben ($S = 16$), aber die Lösung soll auch für andere Werte funktionieren.

> Die Nummer des Teilnehmers, der gerade an der Reihe ist, eine Silbe zu sprechen, nennen wir A (für aktuell). Da Gina die allererste Silbe spricht, ist zu Beginn $A = 0$.

Die Abbildung zeigt, was bei $R = 7$ Teilnehmern passiert, wenn ein Satz mit $S = 4$ Silben gesprochen wird (z.B. „In-for-ma-tik") und dies bei Position $A = 5$ beginnt.



Man sieht, dass die „Silbenposition" um $S - 1 = 3$ Schritte weiter rückt. Am Ende darf der Teilnehmer auf Position 1 zum Buffet. Es bleiben 6 Teilnehmer übrig ($R = 6$), und die Teilnehmer auf den bisherigen Positionen 2 bis 6 bekommen nun die Positionen 1 bis 5. Die nächste „Sprechrunde" startet dann bei $A = 1$.

Damit wir nicht das Sprechen der einzelnen Silben mitzählen müssen, wollen wir den neuen Wert von A direkt berechnen. Es genügt aber nicht, diesen „Zeiger" auf den aktuellen Sprecher einfach auf $A + (S - 1)$ zu erhöhen. Im Beispiel oben wäre das 5 + 3 = 8. Wir müssen berücksichtigen, dass es beim Sprechen über die letzte Position $R - 1$ (im Beispiel: 6) hinaus gehen kann. Das gelingt mit einer „Modulo-Rechnung": Diese gibt den Rest an, der beim Teilen einer ganzen Zahl durch eine andere ganze Zahl bleibt. Der neue Wert von A ist also $(A + (S - 1))$ modulo R; im Beispiel: 8 modulo 7 = 1 (8 geteilt durch 7 ist 1 mit Rest 1).

Gina ist wieder an der Reihe, wenn zum ersten Mal wieder $A = 0$ ist. Wenn danach noch R Teilnehmer übrig sind, konnte Gina Platz $(T - R)$ in der Buffetreihenfolge erreichen.

Zwei Silben

Gina hat ja die Möglichkeit, zwei Silben zu sprechen, wenn sie an der Reihe ist. Wenn sie das tut, verringert sie den anschließenden Wert von A im Vergleich zum Normalfall um 1. Wenn sie C-mal die Chance hat, zwei Silben zu sprechen, kann Sie A also um C verringern. Wenn $A \le C$, kann sie A also auf 0 setzen – und dann hat sie gewonnen. Das bedeutet: Wir müssen bei der Simulation die Anzahl der Chancen C richtig berechnen. Sobald $A \le C$ ist, ist klar, dass Gina genau A mal zwei Silben sprechen muss, um A auf 0 zu setzen und zum Buffet zu kommen. Welche ihrer Chancen sie dazu nutzt, ist egal.

Der Satz „kommt bei Gina vorbei", wenn beim Heraufzählen der Silbenposition (ohne Modulo-Rechnung) die Werte R, $2 \cdot R$, $3 \cdot R$ usw. erreicht werden. Wenn wir die abschließende Silbenposition $A + (S - 1)$ durch R teilen, wissen wir, wie oft das passiert ist. Genau so viele Chancen hat Gina, zwei Silben zu sprechen. Im Beispiel oben hat Gina 8 geteilt durch 7 = 1 Chance.

Umsetzung

Da die Lösung sehr genau beschrieben wurde, ist die Umsetzung in Python schnell zu erklären. Die Funktion simulation simuliert die Buffet-Lotterie und berechnet dabei Ginas Chancen, zwei Silben zu sprechen. Die Funktion hat zwei Parameter **teilnehmer** und **silben**, die den Werten T und S entsprechen. Die Funktion verwendet die Variablen **rest**, **aktuell** und **chancen** für die Werte R, A und C. Die Simulation wird in einer Schleife solange durchgeführt, bis irgendwann **aktuell <= chancen** (also $A \le C$, wie beschrieben).

Beispiele

Die Funktion simulation kann mit den gewünschten Werten in der Python-Shell aufgerufen werden und gibt dann an, wie oft Gina zwei Silben sprechen soll.

```
>>> simulation(28, 16)
Gina bekommt Platz 8, wenn sie 2-mal zwei Silben sagt.
```
Das Ergebnis für die Werte aus der Aufgabenstellung.

```
>>> simulation(15, 16)
Gina bekommt Platz 1, wenn sie 0-mal zwei Silben sagt.
```
Immer wenn $S = T + 1$ ist, ist Gina direkt an der Reihe, ohne etwas zu tun.

```
>>> simulation(10, 45)
Gina bekommt Platz 1, wenn sie 4-mal zwei Silben sagt.
```
Immer wenn $S = C \cdot (T + 1) + 1$ ist, ist Gina direkt an der Reihe, wenn sie alle C Chancen nutzt; hier ist $C = 4$.

```
>>> simulation(280, 19)
Gina bekommt Platz 78, wenn sie 5-mal zwei Silben sagt.
```
Bei diesen Zahlen kommt Gina erst sehr spät zum Buffet.

Quelltext

```
def simulation(teilnehmer, silben):
    rest = teilnehmer
    aktuell = 0 # Gina beginnt und ...
    chancen = 1 # ... hat damit gleich eine Chance,
                # zwei Silben zu sprechen
    gina_satt = False # Gina will noch zum Buffet.

    while (not gina_satt):
        # Ein Satz wird gesprochen,
        # die Werte werden aktualisiert.
        aktuell = aktuell + silben - 1
            # Die Silbenposition rückt weiter.
        chancen = chancen + (aktuell // rest)
            # //: ganzzahlige Division
        aktuell = aktuell % rest # %: modulo-Rechnung
        rest = rest - 1 # Einer durfte zum Buffet.

        # Nun überprüfen wir, ob Gina gewinnen kann:
        if (aktuell <= chancen):
            gina_satt = True # Sie kann!
            print("Gina bekommt Platz ", teilnehmer - rest,
                  ", wenn sie ", aktuell, "-mal zwei Silben sagt.",
                  sep='') # Kein Blank zwischen den Ausgaben.
```