6.

while $|A| > 1$ **do**
  $a \leftarrow$ losowy element z A;
  $A \leftarrow A \setminus \{a\}$
  $b \leftarrow$ losowy element z A;
  $A \leftarrow A \setminus \{b\}$
  $A \leftarrow A \cup \{a - b\}$
output (x mod 2), gdzie x jest elementem ze zbioru A

Zauważmy, że $(a-b) \bmod c = (a \bmod c - b \bmod c) \bmod c$, zatem nie potrzebujemy przechowywać całych liczb, a jedynie ich najmniej znaczący bit (wartość modulo 2 to najmniej znaczący bit liczby). Zauważmy, że stały $(a \bmod c - b \bmod c) \bmod c$ możemy przedstawić jako XOR z tychże liczb, tzn.

a%2	b%2	a%2 ^ b%2	(a%2 - b%2)%2
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

W ten sposób potrzebujemy tylko 1 bitu, żeby przechowywać w nim wartość result z wyniku algorytmu. Wtedy jednak będziemy obliczać $(a[i] \% 2)$ ^ result. Natomiast już wybrani losowi elementów tablicy, możemy traktować po prostu po kolei, bo obliczanie jest łączne.

Algorytm:

```
X = A[0] mod 2
for i ← 1 to |A|:
    x = x ^ (A[i] mod 2)
return x
```