Let $b = (b_1 \ldots, b_s)$ be the $m = (m_1, \ldots, m_s)$-mixed radix representation of n computed from the residue vector $x = (x_1, \ldots, x_s)$ via some program. Thus we have $n \equiv b_j \mod m_j$ for all $j \in \{1, \ldots, s\}$.

In order to formally verify that the output vector b is correct we would need to know n as an (long) integer, that is, the recombined n as a single number.

Since we do not n in this way, we can only provide a criterion to catch some incorrect results. Here's one such criterion. This criterion will accept all correct results, but may fail to detect some incrorrect ones. However, this is very unlikely.

The vector $(b_1 \ldots, b_s)$ satisfies the relation

$$n = b_1 + b_2 M_2 + \cdots b_s M_s$$

where M_j is the product of $m_1, \ldots, m_{j-1}$.

Therefore we have $n \equiv b_1 \mod m_1$ and thus we have $x_1 \equiv b_1 \mod m_1$. Next we have $n \equiv b_1 + b_2 M_2 \mod m_2$ and thus we have $x_2 \equiv b_1 + b_2 M_2 \mod m_2$. More generally, for all $j \in \{1, \ldots, s\}$ we have $x_j \equiv b_1 + b_2 M_2 + \cdots + b_j M_j$.

The following pseudo-C function implements this criterion

```
    int is_probably_correct (s,b,x,m)  for (j=1,j<=s,j++)
M = 1;
S = b_1;
for (k=2,k<=j;k++)
M = (M * m_k-1) % (m_j);
S = (S + (b_k * M) % m_j) % m_j;
if (x_j - S) % m_j != 0  return 1;
return 0;
```