

Verhoeff Checksum Algorithm

Use the following tables in the program.

Multiplication Table (D)

```
int d[10][10] = { {0,1,2,3,4,5,6,7,8,9}, {1,2,3,4,0,6,7,8,9,5}, {2,3,4,0,1,7,8,9,5,6},  
{3,4,0,1,2,8,9,5,6,7}, {4,0,1,2,3,9,5,6,7,8}, {5,9,8,7,6,0,4,3,2,1}, {6,5,9,8,7,1,0,4,3,2},  
{7,6,5,9,8,2,1,0,4,3}, {8,7,6,5,9,3,2,1,0,4}, {9,8,7,6,5,4,3,2,1,0} };
```

Permutation Table (P)

```
int p[8][10] = { {0,1,2,3,4,5,6,7,8,9}, {1,5,7,6,2,8,3,0,9,4}, {5,8,0,3,7,9,6,1,4,2},  
{8,9,1,6,0,4,3,5,2,7}, {9,4,5,3,1,2,6,8,7,0}, {4,2,8,6,5,7,3,9,0,1}, {2,7,9,3,8,0,6,4,1,5},  
{7,0,4,6,9,1,3,2,5,8} };
```

Algorithm Steps

For a 12-digit Aadhaar number:

- Start with $c = 0$.
- Process digits from right to left.
- For each digit:
 $c = d[c][p[\text{position} \% 8][\text{digit}]]$;
- After all digits are processed, the Aadhaar number is valid if $c == 0$.

Function to be Used in Thread 3

```
int verhoeffValidate(char *aadhaar) {  
    int c = 0;  
    int len = strlen(aadhaar);  
    for (int i = 0; i < len; i++) {  
        int digit = aadhaar[len - 1 - i] - '0';  
        c = d[c][p[i % 8][digit]];  
    }  
}
```

```
return (c == 0); }
```