

```

87 Miller-Rabin primality test:
88 using u64 = uint64_t;
89 using u128 = __uint128_t;
90
91 u64 binpower(u64 base, u64 e, u64 mod) {
92     u64 result = 1;
93     base %= mod;
94     while (e) {
95         if (e & 1)
96             result = (u128)result * base % mod;
97         base = (u128)base * base % mod;
98         e >>= 1;
99     }
100     return result;
101 }
102
103 bool check_composite(u64 n, u64 a, u64 d, int s) {
104     u64 x = binpower(a, d, n);
105     if (x == 1 || x == n - 1)
106         return false;
107     for (int r = 1; r < s; r++) {
108         x = (u128)x * x % n;
109         if (x == n - 1)
110             return false;
111     }
112     return true;
113 };
114
115 bool MillerRabin(u64 n) {
116     if (n < 2)
117         return false;
118
119     int r = 0;
120     u64 d = n - 1;
121     while ((d & 1) == 0) {
122         d >>= 1;
123         r++;
124     }
125
126     for (int a : {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37}) {
127         if (n == a)
128             return true;
129         if (check_composite(n, a, d, r))
130             return false;
131     }
132     return true;
133 }

```