```
// Euler_phi_table_complexity_O(n)

#define N 1000005

int lp[N + 1];
int phi[N + 1];
vector<int> pr;

void calc_sieve()
{
    phi[1] = 1;
    for (int i = 2; i <= N; ++i)
    {
        if (lp[i] == 0)
        {
            lp[i] = i;
            phi[i] = i - 1;
            pr.push_back(i);
        }
        else
        {
            //Calculating phi
            if (lp[i] == lp[i / lp[i]])
                phi[i] = phi[i / lp[i]] * lp[i];
            else
                phi[i] = phi[i / lp[i]] * (lp[i] - 1);
        }
        for (int j = 0; j < (int)pr.size() && pr[j] <= lp[i] && i * pr[j]
<= N; ++j)
            lp[i * pr[j]] = pr[j];
    }
}/// O(n)









int main()
{
    long long n;
    calc_sieve();
    int T;
    scanf("%d",&T);
    for( int t=1; t<=T; t++ )
    {
        scanf("%d",&n);
        printf("%d\n",phi[n]);
    }
    return 0;
}
```