

a) Código de la simulación:

```
#include <iostream>
#include <set>
using namespace std;

const int tam = 11;
set<int> secuencia;
int a, c;

int d(int i)
{
    if (i == 0)
        return 0;
    else
        return ( (a * d(i - 1) + c) % tam );
}

void insertar(int key, int *tablaHash, int i, int posPosible)
{
    int pos = key % tam;

    if (tablaHash[posPosible] == -1)
    {
        tablaHash[posPosible] = key;
    }
    else
    {
        int posDoble = (pos + d(i)) % tam;

        if (secuencia.find(posDoble) == secuencia.end())
        {
            secuencia.insert(posDoble);
            insertar(key, tablaHash, i+1, posPosible);
        }
    }
}
```

```

int main()
{
    int key = 0;
    int tablaHash[tam];
    bool correcto = false;

    for (int k=0; k<tam; k++)
        tablaHash[k] = -1;

    for (a=0; a<tam; a++)
        for (c=0; c<tam; c++)
        {
            for (int k=0; k<tam; k++)
            {
                insertar(key, tablaHash, 1, key % tam);
                if (secuencia.size() == tam)
                    correcto = true;
                secuencia.clear();
            }

            if (correcto)
            {
                cout << "a=" << a << " c=" << c << endl;
                correcto = false;
            }

            for (int k=0; k<tam; k++)
                tablaHash[k] = -1;
        }
}

```

- b) Observando los valores válidos que consigue la simulación para diferentes M primos podemos concluir que para cualquier M primo (que es un requisito de la función hash) los valores válidos son a igual a 1 y c igual a cualquier numero mayor que 0 y menor que M.