#include<cstdio>

#include<iostream>

#include<cstring>

#include<string>

#include<algorithm>

using namespace std;

template<class T> int task1(T \*a, T b, int n){ //模板类函数 0为边界返回

    if (!n) return a[n] == b;

    else return a[n] == b | task1(a, b, n - 1); //位运算取或

}

inline void task2(int n){ //一种位运算枚举子集的方法，可用于枚举任意一个二进制集合的子集

    int tmp = (1 << n) - 1;//因为题目需要全枚举到 所以定义最大集合为 2^n - 1，注意位运算优先级最低

    for (int i = tmp; i; i = (i - 1) & tmp) //不断减1， 并用位运算保证一定是子集

    {

        for (int j = 0; j < n; ++j){//输出方案

            if (i & (1 << j)) printf("1"); // 判断当前位是否为1

            else printf("0");

        }

        printf("\n");

    }

}

inline void task2\_1(int i, int n){//把上一种方法改为递归版本

    for (int j = 0; j < n; ++j){//输出方案

        if (i & (1 << j)) printf("1"); // 判断当前位是否为1

        else printf("0");

    }

    printf("\n");

    if (i) task2\_1(i - 1, n);

}

int a[10];

int main()

{

    cout<<task1(a, 0, 9)<<endl;

    // task2(5);

    task2\_1((1 << 5) - 1, 5);

    system("pause");

}