



Universidad Católica  
**San Pablo**

## **PROGRAMA PROFESIONAL**

Ciencias de la Computación

## **TÍTULO DEL TRABAJO**

Tarea - SAT

## **CURSO**

Análisis y Diseño de Algoritmos

### **ALUMNOS**

- Angel Josue Loayza Huarachi

**SEMESTRE: V**

**AÑO: 2022**

“El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo

```

#include <iostream>
#include <vector>
#include <math.h>

using namespace std;

void printMatriz(int filas, int columnas, bool matriz[100][100])
{
    cout << "Matriz con las posibles soluciones: " << endl;
    for (int fil = 0; fil < filas; fil++)
    {
        for (int col = 0; col < columnas; col++)
        {
            cout << matriz[fil][col] << " ";
        }
        cout << endl;
    }
}

bool possibleSolutions(bool matriz[100][100], int nvariables)
{
    int columnas = nvariables;
    int filas = pow(2, nvariables);

    int razon = 0;
    int fil = 0;
    int cont = 1;
    int i = 0, div = 2;

    for (int j = 0; j < columnas; j++)
    {
        i = 0;
        cont = 1;
        razon = filas / div;
        do {
            for (int fil = i; fil < i + razon; fil++)
            {
                if (cont % 2 != 0) {
                    matriz[fil][j] = true;
                }
                else {
                    matriz[fil][j] = false;
                }
            }
            cont++;
            i += razon;
        } while (i < filas);
        div *= 2;
    }

    //printMatriz(filas, columnas, matriz);

    return matriz;
}

bool valueIdentifier(int num, vector<bool> respuestas)
{
    // F = (x1 or x2) and (x2 or x1') and (x1' or x2')
    //int a[] = { 1, 2, -1 };
    //int b[] = { 2, -1, -2 };
    // respuestas = [ verdad verdad ]

    if (num < 0)

```

```

        {
            num = num * -1;
            return !(respuestas[num - 1]);
        }

        return respuestas[num - 1];
    }

bool check(int a[], int b[], int m, vector<bool> respuestas)
{
    int aux = 0;
    for (int i = 0; i < m; i++) //iteraciones de los parentesis -> 3
    {
        // ----- To do... -----
        bool part1;
        part1 = valueIdentifier(a[i], respuestas);
        //cout << "part1: " << part1 << endl;
        bool part2;
        part2 = valueIdentifier(b[i], respuestas);
        //cout << "part2: " << part2 << endl;
        // -----

        if (!(part1 || part2)) // si alguno es falso, tenemos que
cambiar valores de part1 y part2
        {
            return false;
        }
    }
    return true;
}

void printRes(vector<bool> respuestas)
{
    int aux = 1;
    for (auto i = respuestas.begin(); i != respuestas.end(); ++i)
    {
        if (*i == 0)
        {
            cout << "x" << aux << " debe ser: Falso" << endl;
        }
        else if (*i == 1)
        {
            cout << "x" << aux << " debe ser: Verdad" << endl;
        }
        //cout << "x" << aux << " debe ser: " << *i << endl;
        aux++;
    }
}

bool SAT(int n, int m, int a[], int b[])
{
    bool matriz[100][100];
    possibleSolutions(matriz, n);

    vector<bool> respuestas;
    for (int i = 0; i < n; i++)
    {
        respuestas.push_back(0);
    }

    int filas = pow(2, n);
    int columnas = n;
    int aux = 0;

```

```

for (int i = 0; i < filas; i++)
{
    for (int j = 0; j < columnas; j++) //damos valores a respuestas
    {
        respuestas[j] = matriz[i][j];
    }

    //Evaluamos
    if ( check(a, b, m, respuestas) )
    {
        cout << "Si existe solucion" << endl;
        printRes(respuestas);
        return true;
    }
}

cout << "No existe solucion" << endl;
return false;
}

int main()
{
    int n, m;

    // F = (x1 or x2) and (x2 or x1') and (x1' or x2')
    // Existe: Si hay solucion
    // Solucion: x1 = false, x2 = true

    n = 2; // n variables
    m = 4; // n parentesis
    int a[] = { -1,  2, -1, -1 };
    int b[] = {  2, -1, -2, -2 };

    cout << "----- SAT -----" << endl;
    SAT(n, m, a, b);

    cout << endl << "Fin del programa" << endl;
    return 0;
}

```