Sudoku(BT-H-1)

Program to solve the sudoku puzzle by filling the empty cell

Input Format

1. Take a number for the size of Sudoku grid. Size will be NxN

Constraints

1. size > 0
2. num > = 0

Output Format

1. It should find an empty cell and fill it with correct number
2. If Not poosible Print "No Solution Exist!"

Sample Input 0

```
3 0 6 5 0 8 4 0 0
5 2 0 0 0 0 0 0 0
0 8 7 0 0 0 0 3 1
0 0 3 0 1 0 0 8 0
9 0 0 8 6 3 0 0 5
0 5 0 0 9 0 6 0 0
1 3 0 0 0 0 2 5 0
0 0 0 0 0 0 0 7 4
0 0 5 2 0 6 3 0 0
```

Sample Output 0

```
3 1 6 5 7 8 4 9 2
5 2 9 1 3 4 7 6 8
4 8 7 6 2 9 5 3 1
2 6 3 4 1 5 9 8 7
9 7 4 8 6 3 1 2 5
8 5 1 7 9 2 6 4 3
1 3 8 9 4 7 2 5 6
6 9 2 3 5 1 8 7 4
7 4 5 2 8 6 3 1 9
```

Current Buffer (saved locally, editable) C++

```
1 #include<iostream>
2 #include<bits/stdc++.h>
3 #define n 9
4 using namespace std;
5 bool Ans(int s[n][n],int &r,int &c)
6 {
7     for(r=0;r<n;r++)
8     {
9         for(c=0;c<n;c++)
```

```
        {
            if(s[r][c]==0)
            return false;
        }
    }
    return true;
}
bool is_row(int s[n][n],int r,int c,int num)
{
    for(int i=0;i<n;i++)
    {
        if(s[r][i]==num)
        return false;
    }
    return true;

}
bool is_col(int s[n][n],int r,int c,int num)
{
    for(int i=0;i<n;i++)
    {
        if(s[i][c]==num)
        return false;
    }
    return true;

}
bool is_box(int s[n][n],int r,int c,int num)
{
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
        if(s[r+i][c+j]==num)
        return false;
        }
```

```
    }
    return true;

}
bool is_safe(int s[n][n],int r,int c,int num)
{
    return is_row(s,r,c,num)&&is_col(s,r,c,num)&&is_box(s,r-r%3,c-c%3,num);
}
bool solvesuduku(int s[n][n])
{
    int r,c;
    if(Ans(s,r,c))
    return true;
    for(int i=1;i<=n;i++)
    {
        if(is_safe(s,r,c,i))
        {
            s[r][c]=i;
            if(solvesuduku(s))
            return true;
            s[r][c]=0;
        }
    }
    return false;
}
void print(int s[n][n])
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<s[i][j]<<" ";
        }
    }
}
int main()
```

```
{
  int s[n][n];
       for(int j=0;j<n;j++)
       {
           for(int k=0;k<n;k++)
           {
               cin>>s[j][k];
           }
       }
      if(solvesuduku(s))
      {
           for(int j=0;j<n;j++)
       {
           for(int k=0;k<n;k++)
           {
               cout<<s[j][k]<<" ";
           }
                cout<<endl;
       }
      }


   return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
3 0 6 5 0 8 4 0 0
5 2 0 0 0 0 0 0 0
0 8 7 0 0 0 0 3 1
0 0 3 0 1 0 0 8 0
9 0 0 8 6 3 0 0 5
0 5 0 0 9 0 6 0 0
1 3 0 0 0 0 2 5 0
0 0 0 0 0 0 0 7 4
0 0 5 2 0 6 3 0 0
```

Your Output (stdout)

```
3 1 6 5 7 8 4 9 2
5 2 9 1 3 4 7 6 8
4 8 7 6 2 9 5 3 1
2 6 3 4 1 5 9 8 7
9 7 4 8 6 3 1 2 5
8 5 1 7 9 2 6 4 3
1 3 8 9 4 7 2 5 6
6 9 2 3 5 1 8 7 4
7 4 5 2 8 6 3 1 9
```

Expected Output

```
3 1 6 5 7 8 4 9 2
5 2 9 1 3 4 7 6 8
4 8 7 6 2 9 5 3 1
2 6 3 4 1 5 9 8 7
9 7 4 8 6 3 1 2 5
8 5 1 7 9 2 6 4 3
1 3 8 9 4 7 2 5 6
6 9 2 3 5 1 8 7 4
7 4 5 2 8 6 3 1 9
```