



PRACTICE

CERTIFICATION^{NEW}

COMPETE

JOBS

LEADERBOARD



Search



swatantragoswam1

[All Contests](#) > [PRP_code](#) > [Sudoku\(BT-H-1\)](#)

Sudoku(BT-H-1)

Problem

Submissions

Leaderboard

Discussions

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 possible Print "No Solution Exist!"

Sample Input 0



Submissions: 0

Max Score: 30

Difficulty: Medium



Rate This Challenge:

[More](#)

```
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++)
```

```
10     {
11         if(s[r][c]==0)
12             return false;
13     }
14 }
15 return true;
16 }
17 bool is_row(int s[n][n],int r,int c,int num)
18 {
19     for(int i=0;i<n;i++)
20     {
21         if(s[r][i]==num)
22             return false;
23     }
24     return true;
25 }
26 }
27 bool is_col(int s[n][n],int r,int c,int num)
28 {
29     for(int i=0;i<n;i++)
30     {
31         if(s[i][c]==num)
32             return false;
33     }
34     return true;
35 }
36 }
37 bool is_box(int s[n][n],int r,int c,int num)
38 {
39     for(int i=0;i<3;i++)
40     {
41         for(int j=0;j<3;j++)
42         {
43             if(s[r+i][c+j]==num)
44                 return false;
45         }
```

```
46     }
47     return true;
48
49 }
50 bool is_safe(int s[n][n],int r,int c,int num)
51 {
52     return is_row(s,r,c,num)&&is_col(s,r,c,num)&&is_box(s,r-r%3,c-c%3,num);
53 }
54 bool solvesudoku(int s[n][n])
55 {
56     int r,c;
57     if(Ans(s,r,c))
58         return true;
59     for(int i=1;i<=n;i++)
60     {
61         if(is_safe(s,r,c,i))
62         {
63             s[r][c]=i;
64             if(solvesudoku(s))
65                 return true;
66             s[r][c]=0;
67         }
68     }
69     return false;
70 }
71 void print(int s[n][n])
72 {
73     for(int i=0;i<n;i++)
74     {
75         for(int j=0;j<n;j++)
76         {
77             cout<<s[i][j]<<" ";
78         }
79     }
80 }
81 int main()
```

```
82  {
83      int s[n][n];
84      for(int j=0;j<n;j++)
85      {
86          for(int k=0;k<n;k++)
87          {
88              cin>>s[j][k];
89          }
90      }
91      if(solvesudoku(s))
92      {
93          for(int j=0;j<n;j++)
94          {
95              for(int k=0;k<n;k++)
96              {
97                  cout<<s[j][k]<<" ";
98              }
99              cout<<endl;
100          }
101      }
102
103
104      return 0;
105  }
```

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
```

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)