



CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of CSE

Implementation of Defective Chess problem

Course no.: CSE-244

Course title: Algorithm Design and Analysis (Sessional)

Experiment no.: 1

Date of experiment: 27/11/2018

Date of submission: 3/12/2018

REMARKS

Name: Ashish Pondit

ID: 1604071

Section: B

Group: B1

Experiment name:

Implementation of Defective Chess problem for the given problem.

Problem Statement:

Suppose you have to arrange a programming contest. The team number is given to you. Each team consists of three coders. Since they are a team so they must sit together. So the three coders must form an “L” shape to do this. The shape can be rotated. Each team will need three desktop.

You are provided with a lab. But the (m,n) the index desktop is defected. You can't move the desktop.

Now arrange the team members in such a way that they can seat together.

Here lab size must be in $2^K * 2^K$ form.

Input :

You have to take the team no “n” and the index of the defected desktop “m” ”n”.

Output :

You have to print the arrangement of the team.

Source code:

```
#include <iostream>
#include <iomanip>
#include <stdlib.h>
using namespace std;
int Board[200][200], siz, tile = 1,temp;

void TileBoard(int tr, int tc, int dr, int dc, int siz)
```

[illegible]

```

// 33333333333333333333333333333333
if (dr >= tr + s && dc < tc + s)
    TileBoard(tr+s, tc, dr, dc, s);
else
{
    Board[tr + s][tc + s - 1] = t;
    TileBoard(tr+s, tc, tr+s, tc+s-1, s);
}

// 44444444444444444444444444444444
if (dr >= tr + s && dc >= tc + s)
    TileBoard(tr+s, tc+s, dr, dc, s);
else
{
    Board[tr + s][tc + s] = t;
    TileBoard(tr+s, tc+s, tr+s, tc+s, s);
}
}

```

```

void OutputBoard(int siz)
{
    for (int i = 0; i < siz; i++)
    {
        for (int j = 0; j < siz; j++)

```

```

    {
        if(Board[i][j]<=temp)
            cout << setw (5) << Board[i][j];
        else
            cout << setw (5) <<"0";
    }
    cout << endl;
    cout << endl;
}
}

int main()
{
    int k, dr, dc,i,t=0;
    cout << "Enter how many team to perticipate" << endl;
    cin >> temp;
    for(i=1; t<temp; i=i*2)
    {
        t=(i*i-1)/3;
        if(t>=temp)
            break;
    }
    siz=i;
    cout << "Enter location of defect computer" << endl;
    cin >> dr >> dc;

```

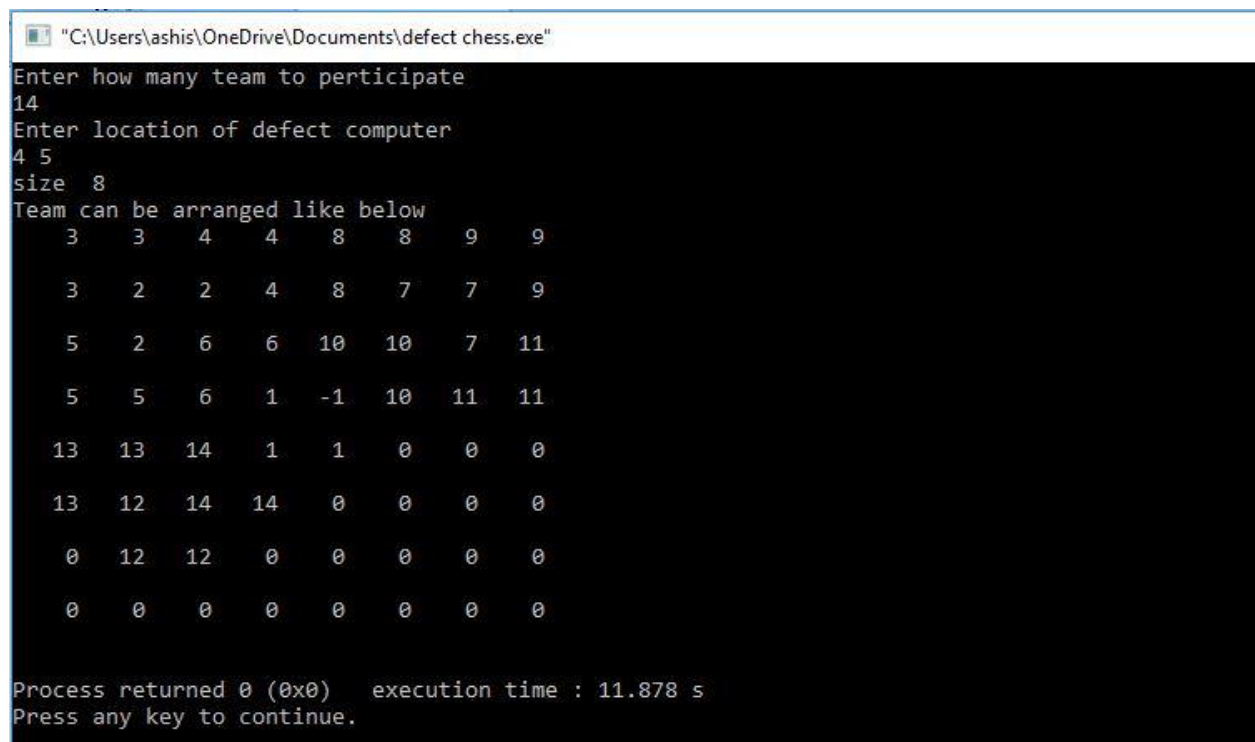
```

dr--;
dc--;
Board[dr][dc] = -1;
cout<<"size  "<<siz<<endl;
TileBoard(0, 0, dr, dc, siz);
cout<<"Team can be arranged like below"<<endl;
OutputBoard(siz);
}

```

Sample Input/Output:

The sample input and output is given below.



```

"C:\Users\ashis\OneDrive\Documents\defect chess.exe"
Enter how many team to participate
14
Enter location of defect computer
4 5
size 8
Team can be arranged like below
 3   3   4   4   8   8   9   9
 3   2   2   4   8   7   7   9
 5   2   6   6  10  10   7  11
 5   5   6   1  -1  10  11  11
13  13  14   1   1   0   0   0
13  12  14  14   0   0   0   0
 0  12  12   0   0   0   0   0
 0   0   0   0   0   0   0   0

Process returned 0 (0x0)   execution time : 11.878 s
Press any key to continue.

```

Analysis of code:

Here we took the number of teams and using the formula $(2^K * 2^K - 1)/3$ we found out that we need 8*8 capacity lab to arrange the team.

After that we applied defective chess method to fill up the tem . When the team number is exceeded our work is done and I printed the rest of the seat to Zero.