asicstyle=

1 TIC TAC TOE PROGRAM By ANIKET BANDI

2 Aim

The aim of this project is to develop a tic tac toe game. The game consists of 2 parts, one a single player game(a player against a system) and other a multilplayer game(two players on their devices playing against each other).

3 Code in C++ Language:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
                char ch;
                do{
                        system("CLS");
                        int i=0;
        string arr[3][3]={{"1","2","3"},{"4","5","6"},{"7","8","9"}};
do{
    cout<<endl<<endl;
    int a;
     cout<<arr[0][0]<<" | "<<arr[0][1]<<" | "<<arr[0][2]<<endl;
     cout<<"___|____|___"<<endl;
     cout<<arr[1][0]<<" | "<<arr[1][1]<<" | "<<arr[1][2]<<endl;
     cout<<"___|____|___"<<endl;
     cout<<arr[2][0]<<" | "<<arr[2][1]<<" | "<<arr[2][2]<<endl;
     cout<<"   |    |   "<<endl;
     cout<<"\nEnter player A"<<endl;
    cin>>a;
switch(a)
{
        case 1:
                if(arr[0][0]!="A"&&arr[0][0]!="B")
                {
                arr[0][0]="A";
            }
            else{
                    cout<<"Already Filled try in next turn"<<endl;
                }
                break;
        case 2:
                if(arr[0][1]!="A"&&arr[0][1]!="B")
                {
                arr[0][1]="A";
            }
        else{
```

```
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
    case 3:
            if(arr[0][2]!="A"&&arr[0][2]!="B")
            {
            arr[0][2]="A";
        }
            else{
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
    case 4:
            if(arr[1][0]!="A"&&arr[1][0]!="B")
            {
            arr[1][0]="A";
    }
    else{
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
    case 5:
            if(arr[1][1]!="A"&&arr[1][1]!="B")
            {
            arr[1][1]="A";
    }
            else{
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
    case 6:
            if(arr[1][2]!="A"&&arr[1][2]!="B")
            {
            arr[1][2]="A";
            }
            else{
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
    case 7:
            if(arr[2][0]!="A"&&arr[2][0]!="B")
            {
            arr[2][0]="A";
            }
            else{
                cout<<"Already Filled try in next turn"<<endl;
            }
            break;
```

```
        case 8:
                if(arr[2][1]!="A"&&arr[2][1]!="B")
                {
                arr[2][1]="A";
        }
                else{
                    cout<<"Already Filled try in next turn"<<endl;
                }
                break;
        case 9:
                        if(arr[2][2]!="A"&&arr[2][2]!="B")
                {
                arr[2][2]="A";
                }
                else{
                    cout<<"Already Filled try in next turn"<<endl;
                }
                break;
        default:
                        cout<<"Invalid input please try again in next turn"<<endl;


}
if(arr[0][0]==arr[0][1]&&arr[0][1]==arr[0][2]||arr[1][0]==arr[1][1]&&arr[1][1]==arr[1]
||arr[0][0]==arr[1][0]&&arr[1][0]==arr[2][0]||arr[0][1]==arr[1][1]&&arr[1][1]==arr[2][
||arr[0][0]==arr[1][1]&&arr[1][1]==arr[2][2]||arr[0][2]==arr[1][1]&&arr[1][1]==arr[2][
{
                    cout<<"**************Congratulations**************\n_________Play
        break;
}
                 cout<<"\nEnter player B"<<endl;
                 cin>>a;
switch(a)
{
        case 1:
                if(arr[0][0]!="A"&&arr[0][0]!="B")
                {
                arr[0][0]="B";
            }
            else{
                    cout<<"Already Filled try in next turn"<<endl;

                }
                break;
        case 2:
                if(arr[0][1]!="A"&&arr[0][1]!="B")
                {
                arr[0][1]="B";
```

```
    }
else{
            cout<<"Already Filled try in next turn"<<endl;

        }
        break;
case 3:
        if(arr[0][2]!="A"&&arr[0][2]!="B")
        {
        arr[0][2]="B";
    }
        else{
            cout<<"Already Filled try in next turn"<<endl;

        }
        break;
case 4:
        if(arr[1][0]!="A"&&arr[1][0]!="B")
        {
        arr[1][0]="B";
}
else{
            cout<<"Already Filled try in next turn"<<endl;

        }
        break;
case 5:
        if(arr[1][1]!="A"&&arr[1][1]!="B")
        {
        arr[1][1]="B";
}
        else{
            cout<<"Already Filled try in next turn"<<endl;

        }
        break;
case 6:
        if(arr[1][2]!="A"&&arr[1][2]!="B")
        {
        arr[1][2]="B";
        }
        else{
            cout<<"Already Filled try in next turn"<<endl;
        }
        break;
case 7:
        if(arr[2][0]!="A"&&arr[2][0]!="B")
        {
```

```
                arr[2][0]="B";
                }
                else{
                    cout<<"Already Filled try in next turn"<<endl;

                }
                break;
        case 8:
                if(arr[2][1]!="A"&&arr[2][1]!="B")
                {
                arr[2][1]="B";
        }
                else{
                    cout<<"Already Filled try in next turn"<<endl;

                }
                break;
        case 9:
                        if(arr[2][2]!="A"&&arr[2][2]!="B")
                {
                arr[2][2]="B";
                }
                else{
                    cout<<"Already Filled try in next turn"<<endl;

                }
                break;
        default :
                        cout<<"Invalid input please try again in next turn"<<endl;
}

if(arr[0][0]==arr[0][1]&&arr[0][1]==arr[0][2]||arr[1][0]==arr[1][1]&&arr[1][1]==arr[1]
||arr[0][0]==arr[1][0]&&arr[1][0]==arr[2][0]||arr[0][1]==arr[1][1]&&arr[1][1]==arr[2][
||arr[0][0]==arr[1][1]&&arr[1][1]==arr[2][2]||arr[0][2]==arr[1][1]&&arr[1][1]==arr[2][
{
                cout<<"Congratulations\n         Player B won the Game"<<endl;
        break;
}

system("CLS");
i++;
}while(i<5);

    cout<<"Press Y if you want to contine... "<<endl;
    cin>>ch;
}while(ch=='y'||ch=='Y');

return 0;
```

```
}
```

4 C++ output

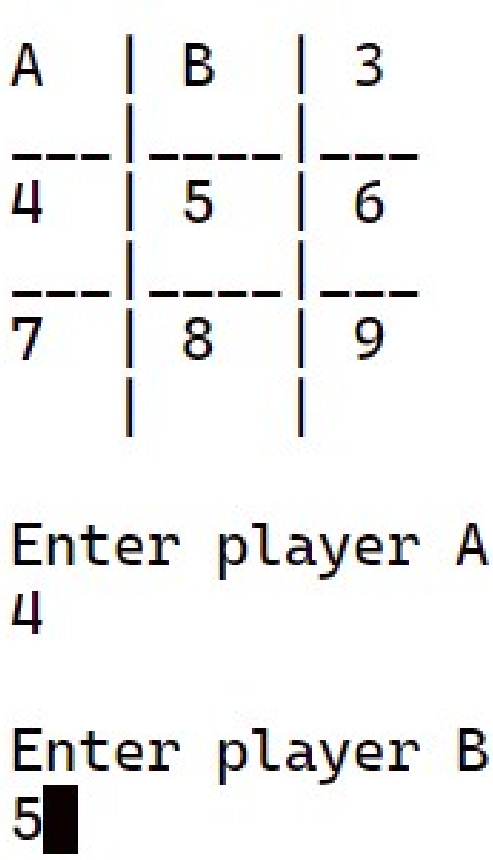

Figure 1: OUTPUT 1

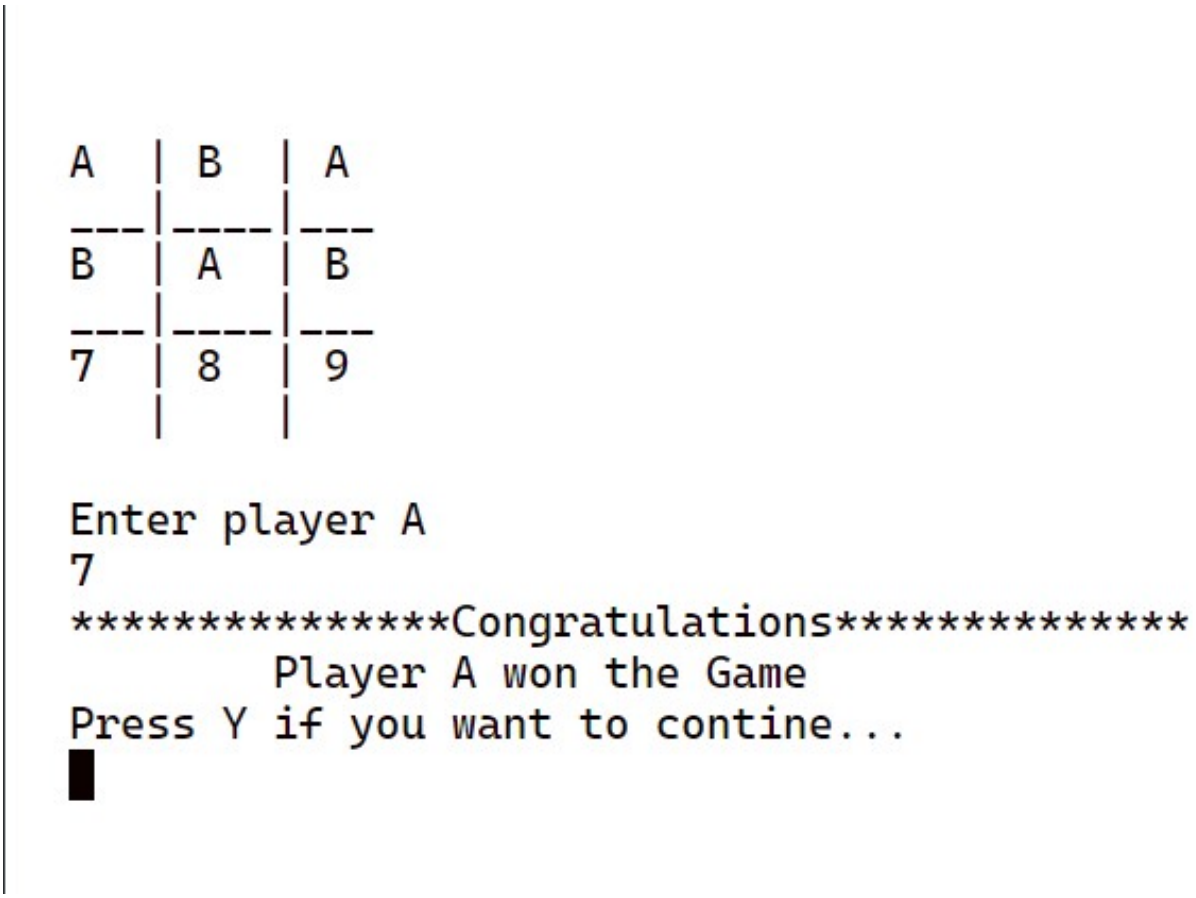

Figure 2: OUTPUT 2

5 C++ Profiling Output:

```
hp@SJ MINGW64 ~/sj/aniket (master)
$ gprof a.exe gmon.out
BFD: Dwarf Error: Could not find abbrev number 108.
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00       60     0.00     0.00  std::char_traits<char>::compare(char const*, char const*, unsigned
int)
  0.00      0.00     0.00       60     0.00     0.00  __gnu_cxx::__enable_if<std::__is_char<char>::__value, bool>::__type
 std::operator==<char>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, std::__cxx
11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&)
  0.00      0.00     0.00       14     0.00     0.00  bool std::operator==<char, std::char_traits<char>, std::allocator<c
har> >(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*)
  0.00      0.00     0.00       14     0.00     0.00  bool std::operator!=<char, std::char_traits<char>, std::allocator<c
har> >(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*)

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
           else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
           function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.


Copyright (C) 2012-2018 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


                     Call graph (explanation follows)
```

```
granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00      60/60          __gnu_cxx::__enable_if<std::__is_char<char>::__value, bool>::__type std:
:operator==<char>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, std::__cxx11::b
asic_string<char, std::char_traits<char>, std::allocator<char> > const&) [3]
[2]      0.0    0.00    0.00      60         std::char_traits<char>::compare(char const*, char const*, unsigned int) [2]
-----------------------------------------------
                0.00    0.00      26/60          _fu29___ZSt4cout [95]
                0.00    0.00      34/60          _fu17___ZSt4cout [82]
[3]      0.0    0.00    0.00      60         __gnu_cxx::__enable_if<std::__is_char<char>::__value, bool>::__type std::ope
rator==<char>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, std::__cxx11::basic
_string<char, std::char_traits<char>, std::allocator<char> > const&) [3]
                0.00    0.00      60/60          std::char_traits<char>::compare(char const*, char const*, unsigned int)
[2]
-----------------------------------------------
                0.00    0.00      14/14          bool std::operator!=<char, std::char_traits<char>, std::allocator<char>
>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*) [5]
[4]      0.0    0.00    0.00      14         bool std::operator==<char, std::char_traits<char>, std::allocator<char> >(st
d::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*) [4]
-----------------------------------------------
                0.00    0.00       2/14          _fu32___ZSt3cin [99]
                0.00    0.00       2/14          _fu9___ZSt4cout [108]
                0.00    0.00       2/14          _fu11___ZSt4cout [76]
                0.00    0.00       2/14          _fu15___ZSt4cout [80]
                0.00    0.00       2/14          _fu20___ZSt4cout [86]
                0.00    0.00       2/14          _fu22___ZSt4cout [88]
                0.00    0.00       2/14          _fu25___ZSt4cout [91]
[5]      0.0    0.00    0.00      14         bool std::operator!=<char, std::char_traits<char>, std::allocator<char> >(st
d::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*) [5]
                0.00    0.00      14/14          bool std::operator==<char, std::char_traits<char>, std::allocator<char>
>(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*) [4]
-----------------------------------------------

 This table describes the call tree of the program, and was sorted by
 the total amount of time spent in each function and its children.

 Each entry in this table consists of several lines.  The line with the
 index number at the left hand margin lists the current function.
 The lines above it list the functions that called this function,
 and the lines below it list the functions this one called.
 This line lists:
     index      A unique number given to each element of the table.
                Index numbers are sorted numerically.
                The index number is printed next to every function name so
                it is easier to look up where the function is in the table.

     % time     This is the percentage of the `total' time that was spent
                in this function and its children.  Note that due to
                different viewpoints, functions excluded by options, etc,
                these numbers will NOT add up to 100%.

     self       This is the total amount of time spent in this function.

     children   This is the total amount of time propagated into this
                function by its children.

     called     This is the number of times the function was called.
```

```
                If the function called itself recursively, the number
                only includes non-recursive calls, and is followed by
                a `+' and the number of recursive calls.

    name        The name of the current function.  The index number is
                printed after it.  If the function is a member of a
                cycle, the cycle number is printed between the
                function's name and the index number.


For the function's parents, the fields have the following meanings:

    self        This is the amount of time that was propagated directly
                from the function into this parent.

    children    This is the amount of time that was propagated from
                the function's children into this parent.

    called      This is the number of times this parent called the
                function `/' the total number of times the function
                was called.  Recursive calls to the function are not
                included in the number after the `/'.

    name        This is the name of the parent.  The parent's index
                number is printed after it.  If the parent is a
                member of a cycle, the cycle number is printed between
                the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

    self        This is the amount of time that was propagated directly
                from the child into the function.

    children    This is the amount of time that was propagated from the
                child's children to the function.

    called      This is the number of times the function called
                this child `/' the total number of times the child
                was called.  Recursive calls by the child are not
                listed in the number after the `/'.

    name        This is the name of the child.  The child's index
                number is printed after it.  If the child is a
                member of a cycle, the cycle number is printed
                between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole.  This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.
```

```
If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.

For the function's children, the fields have the following meanings:

    self       This is the amount of time that was propagated directly
               from the child into the function.

    children   This is the amount of time that was propagated from the
               child's children to the function.

    called     This is the number of times the function called
               this child `/' the total number of times the child
               was called.  Recursive calls by the child are not
               listed in the number after the `/'.

    name       This is the name of the child.  The child's index
               number is printed after it.  If the child is a
               member of a cycle, the cycle number is printed
               between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole.  This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.


Copyright (C) 2012-2018 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.


Index by function name

   [2] std::char_traits<char>::compare(char const*, char const*, unsigned int) [4] bool std::operator==<char, std::char_t
raits<char>, std::allocator<char> >(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const
&, char const*)
   [3] __gnu_cxx::__enable_if<std::__is_char<char>::__value, bool>::__type std::operator==<char>(std::__cxx11::basic_stri
ng<char, std::char_traits<char>, std::allocator<char> > const&, std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> > const&) [5] bool std::operator!=<char, std::char_traits<char>, std::allocator<char> >(std::__cxx11
::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, char const*)
```

6 C++ Debugging Output:

```
PS G:\Mini_Project> cd "c:\Users\hp\sj\aniket\" ; if ($?) { g++ aniket.cpp -o ani
aniket.cpp: In function 'int main()':
aniket.cpp:127:52: error: expected ';' before '}' token
       cout<<"Already Filled try in next turn"<<endl
                                                    ^
                                                    ;
aniket.cpp:129:3:
   }
   ~
PS C:\Users\hp\sj\aniket>
```

Figure 3: INITIAL WRONG OUTPUT ON terminal

```
[Inferior 1 (process 75688) exited normally]
(gdb) break 127
Breakpoint 1 at 0x403cb8: file ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) break main
Breakpoint 2 at 0x4015d1
(gdb) info breakpoints
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x00403cb8 ../../../../../src/gcc-8.1.0/libgcc/config/i
2       breakpoint     keep y   0x004015d1 <main+17>
(gdb) run
Starting program: C:\Users\hp\sj\aniket\a.exe
[New Thread 65992.0x18c14]
[New Thread 65992.0x44dc]
[New Thread 65992.0x1048c]
[New Thread 65992.0xf074]
[New Thread 65992.0xa768]

Thread 1 hit Breakpoint 1, __chkstk_ms ()
    at ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S:143
143     ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S: No such file or direc
(gdb)
```

Figure 4: Program stop while running in terminal at breakpoints

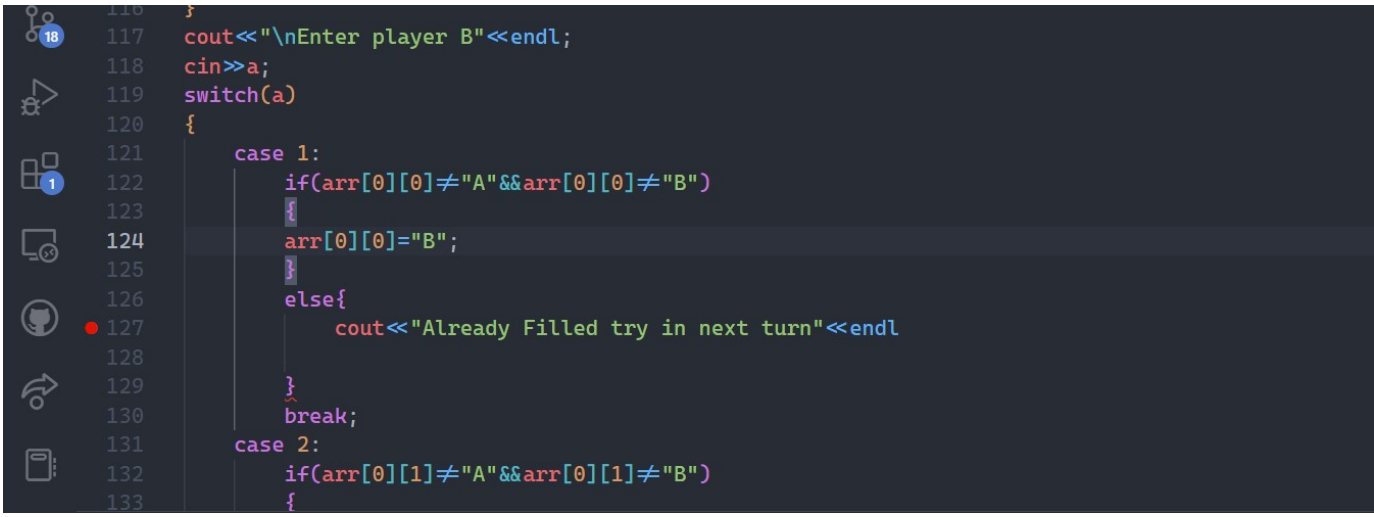

Figure 5: mistake in code

```
(gdb) run[args]
The program being debugged has been started already.
Start it from the beginning? (y or n) [answered Y; input not from terminal]
Starting program: C:\Users\hp\sj\aniket\a.exe [args]
[New Thread 15024.0x1761c]
[New Thread 15024.0xcea4]
[New Thread 15024.0x3cf4]
[New Thread 15024.0x16cb4]
[New Thread 15024.0x77d0]

Thread 1 hit Breakpoint 1, __chkstk_ms ()
    at ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S:143
143     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) info b
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x00403cb8 ../../../../../src/gcc-8.1.0/libgc
        breakpoint already hit 1 time
2       breakpoint     keep y   0x004015d1 <main+17>
(gdb) n
145     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
147     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
148     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
149     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
157     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
158     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
160     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
162     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
164     in ../../../../../src/gcc-8.1.0/libgcc/config/i386/cygwin.S
(gdb) n
0x004032e1 in _pei386_runtime_relocator ()
(gdb) n
Single stepping until exit from function _pei386_runtime_relocator,
which has no line number information.
```

Figure 6: debugging in terminal

```
(gdb) run
Starting program: C:\Users\hp\sj\aniket\a.exe
[New Thread 97036.0xc8c]
[New Thread 97036.0x17698]
[New Thread 97036.0x8a58]
[New Thread 97036.0x114cc]
[New Thread 97036.0xc544]
[New Thread 97036.0x3360]
[New Thread 97036.0xa7c4]
[New Thread 97036.0x954]



1   | 2   | 3
____|_____|____
4   | 5   | 6
____|_____|____
7   | 8   | 9
    |     |

Enter player A
```

Figure 7: code running without error

```
1   | 2   | 3
____|_____|____
4   | 5   | 6
____|_____|____
7   | 8   | 9
    |     |

Enter player A
1

Enter player B
2


A   | B   | 3
____|_____|____
4   | 5   | 6
____|_____|____
7   | 8   | 9
    |     |

Enter player A
3

Enter player B
4


A   | B   | A
____|_____|____
B   | 5   | 6
____|_____|____
7   | 8   | 9
    |     |

Enter player A
5

Enter player B
6


A   | B   | A
    |     |
```