

## הגשת תרגיל

### קוד התרגיל: EX6

הכניסו כאן קישור-שיבוט למאגר שלכם בגיטהאב (חייב להתחיל ב `https://` ולהסתיים ב `.git`):

<input type="text"/>	ת"ז 1:
<input type="text"/>	ת"ז 2:
<input type="text"/>	ת"ז 3:

שמות סטודנטים:

### תוצאות ההגשה

```
> {"exercise":"EX6","git_url":"https://github.com/sholmiw/cpp-course-tic-tac-toe-tesk.git","ID_1":"","ID_2":"","ID_3":"","student_names":""}
< Submission starting!
> {"exercise":"EX6","git_url":"https://github.com/sholmiw/cpp-course-tic-tac-toe-tesk.git","ID_1":"","ID_2":"","ID_3":"","student_names":""}
< ! cd sholmiw
< ! git clone https://github.com/sholmiw/cpp-course-tic-tac-toe-tesk.git
< Cloning into 'cpp-course-tic-tac-toe-tesk'...
< ! cd cpp-course-tic-tac-toe-tesk
< ! grade
< clang++-5.0 --compile Board.cpp -o Board.o
< running test- main.cpp
<
< main.cpp compiled
<
<
~~~~~
< a.out is a Catch v2.2.2 host application.
< Run with -? for options
<
< -----
-----
< constraction
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
<
< -----
-----
< main.cpp:18
< .....
<
< main.cpp:20:
< PASSED:
```

```

< CHECK( BoardToString(board1) == emptyBoard )
< with expansion:
< "....
< ....
< ....
< ....
< "
< ==
< "....
< ....
< ....
< ....
< "
<
< -----
----
< constraction
< -----
----
< main.cpp:14
< .....
<
< main.cpp:24:
< PASSED:
< CHECK( BoardToString(board1) == BoardToString(board2) )
< with expansion:
< "....
< ....
< ....
< ....
< "
< ==
< "....
< ....
< ....
< ....
< "
<
< -----
----
< operator[]1
< -----
----
< main.cpp:27
< .....
<
< main.cpp:32:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:35:
< PASSED:

```

```

< CHECK( BoardToString(board1) == expectedBoard2 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....
< .....O
< "
<
< main.cpp:38:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard3 )
< with expansion:
< "O.....
< .....
< .....
< .....
< .....
< .....O
< "
< ==
< "O.....
< .....
< .....
< .....
< .....
< .....O
< "
<
< main.cpp:41:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard4 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
<
< -----
< ----
< operator[]2
< -----
< ----
< main.cpp:43
< .....
<
< main.cpp:48:
< PASSED:

```

```

< CHECK( board1[{5 , 5}] == 'X' )
< with expansion:
< X == 'X'
<
< -----
----
< operator[]3
< -----
----
< main.cpp:57
< .....
<
< main.cpp:62:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5,5}]='1'; }() )
<
< main.cpp:65:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = '0'; }() )
<
< main.cpp:68:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'x'; }() )
<
< main.cpp:71:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'o'; }() )
<
< main.cpp:74:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....
< "
<
< -----
----
< operator[]4
< -----
----
< main.cpp:92
< .....
<
< main.cpp:97:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 10}] = 'X'; }() )
<
< main.cpp:100:
< PASSED:
< CHECK_THROWS( [&]() { board1[{15, 5}] = 'O'; }() )
<
< main.cpp:103:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:

```

```

< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....
< "
<
< -----
----
< operator[]5
< -----
----
< main.cpp:122
< .....
<
< main.cpp:127:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 10}] = 'M'; }() )
<
< main.cpp:130:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'I'; }() )
<
< main.cpp:132:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....
< "
<
< -----
----
< operator=
< -----
----
< main.cpp:135
< .....
<
< main.cpp:140:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....

```

```

< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:143:
< PASSED:
< CHECK( BoardToString(board2) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:147:
< PASSED:
< CHECK( BoardToString(board3) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:151:
< PASSED:
< CHECK( BoardToString(board4) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....

```

```

< .....
< .....X
< "
<
< main.cpp:157:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:159:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:161:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< -----
< ----
< copy ctor

```

```

< -----
< ---
< main.cpp:164
< .....
<
< main.cpp:172:
< PASSED:
< CHECK( BoardToString(board2) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:176:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
<
=====
< All tests passed (27 assertions in 8 test cases)
<
< ==1352== definitely lost: 0 bytes in 0 blocks
< ==1352== indirectly lost: 0 bytes in 0 blocks
< ==1352== possibly lost: 0 bytes in 0 blocks
< saving grades
< 0
< total =55.0, passed = 55.0
< your grade is :100.0
< raw results
<
<
~~~~~
< a.out is a Catch v2.2.2 host application.
< Run with -? for options
<
< -----
< ---
< constraction
< Please use endl (\\n) at the end of the last line
< Please use endl (\\n) at the end of the last line

```



```

< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
< Please use endl\n at the end of the last line
<
< -----
< ---
< main.cpp:18
< .....
<
< main.cpp:20:
< PASSED:
< CHECK( BoardToString(board1) == emptyBoard )
< with expansion:
< "....
< ....
< ....
< ....
< "
< ==
< "....
< ....
< ....
< ....
< "
<
< -----
< ---
< constra\n
< -----
< ---
< main.cpp:14
< .....
<
< main.cpp:24:
< PASSED:
< CHECK( BoardToString(board1) == BoardToString(board2) )
< with expansion:
< "....
< ....
< ....
< ....
< "
< ==
< "....
< ....
< ....
< ....
< "
<
< -----
< ---
< operator[]1
< -----
< ---
< main.cpp:27
< .....

```

```

<
< main.cpp:32:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:35:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard2 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
<
< main.cpp:38:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard3 )
< with expansion:
< "O.....
< .....
< .....
< .....
< .....
< .....O
< "
< ==
< "O.....
< .....
< .....
< .....
< .....
< .....O
< "
<
< main.cpp:41:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard4 )
< with expansion:
< ".....
< .....
< .....

```

```

< .....
< .....
< .....O
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....O
< "
<
< -----
-----
< operator[]2
< -----
-----
< main.cpp:43
< .....
<
< main.cpp:48:
< PASSED:
< CHECK( board1[{5 , 5}] == 'X' )
< with expansion:
< X == 'X'
<
< -----
-----
< operator[]3
< -----
-----
< main.cpp:57
< .....
<
< main.cpp:62:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5,5}]='1'; }() )
<
< main.cpp:65:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = '0'; }() )
<
< main.cpp:68:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'x'; }() )
<
< main.cpp:71:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'o'; }() )
<
< main.cpp:74:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....
< .....
< .....
< .....
< .....

```

```

< .....
< "
<
< -----
----
< operator[]4
< -----
----
< main.cpp:92
< .....
<
< main.cpp:97:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 10}] = 'X'; }() )
<
< main.cpp:100:
< PASSED:
< CHECK_THROWS( [&]() { board1[{15, 5}] = 'O'; }() )
<
< main.cpp:103:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....
< "
<
< -----
----
< operator[]5
< -----
----
< main.cpp:122
< .....
<
< main.cpp:127:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 10}] = 'M'; }() )
<
< main.cpp:130:
< PASSED:
< CHECK_THROWS( [&]() { board1[{5, 5}] = 'I'; }() )
<
< main.cpp:132:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard1 )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....
< "
< ==
< ".....

```

```

< .....
< .....
< .....
< .....
< .....
< "
<
< -----
----
< operator=
< -----
----
< main.cpp:135
< .....
<
< main.cpp:140:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:143:
< PASSED:
< CHECK( BoardToString(board2) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:147:
< PASSED:
< CHECK( BoardToString(board3) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....

```

```

< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:151:
< PASSED:
< CHECK( BoardToString(board4) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:157:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:159:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:161:

```

```

< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< -----
----
< copy ctor
< -----
----
< main.cpp:164
< .....
<
< main.cpp:172:
< PASSED:
< CHECK( BoardToString(board2) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<
< main.cpp:176:
< PASSED:
< CHECK( BoardToString(board1) == expectedBoard )
< with expansion:
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
< ==
< ".....
< .....
< .....
< .....
< .....
< .....X
< "
<

```

```
<
=====
< All tests passed (27 assertions in 8 test cases)
<
< ==1352== definitely lost: 0 bytes in 0 blocks
< ==1352== indirectly lost: 0 bytes in 0 blocks
< ==1352== possibly lost: 0 bytes in 0 blocks
< compilation notes
< main.cpp compilation notes :
< finished
< We recorded your IDs: , , and your grade: 100.0 date of submission is Wed
May 9 21:47:26 2018
< Connection closed abnormally! Code=1005. Reason=No status code was actually
present.
```