Abcedi Ilacas138180211 ailacas1@mySeneca.ca  
Click or tap here to enter text.

**See the topic's slides, the activity instructions, and the Programming Test Cases.docx**

The number of rows in the tables below are for convenience; they do not indicate the number of cases expected.

**Test Cases for the Black box program**

| **Description** | **+ / − Purpose** | **Data Input** | **Expected Output** | **Actual output if unexpected** | **Success?** | **Comments** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **String: example** |  |  |  |  |
| Extract first character | + | Position: 1 | e | a | No | Incorrect character extracted, possible off-by-one error. |
| Extract middle character | + | Position: 4 | m | l | No | Incorrect character extracted, index possibly misaligned. |
| Extract last character | + | Position: 7 | e | ‘ ‘ | No | Output empty where a character was expected, suggests string is not null terminated properly. |
| Extract at position zero | - | Position: 0 | Error or no output | x | No | Output unexpected character, program may be using 0-indexing unexpectedly. |
| Extract at negative position | - | Position: -1 | Error or no output | e | No | Unexpected output, negative index should not return a valid character. |
|  |  | **String:** **a1b2c3d4e5f6g7!@#** |  |  |  |  |
| Extract first character | + | Position: 1 | a | b | No | The character extracted does not match the expected output. |
| Extract beyond string length | - | Position: 18 | Error or no output | ‘ ‘ | Yes | Correct behavior for out-of-bounds position. |
| Extract last character | + | Position: 17 | # | ‘ ‘ | No | The character extracted does not match the expected output. |
| Extract at position zero | - | Position: 0 | Error or no output | 1 | No | The character extracted should not be valid for position zero. |
| Extract at negative position | - | Position: -1 | Error or no output | a | No | Negative positions should not return valid characters. |
|  |  | **String: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa** |  |  |  |  |
| Extract the first character | + | Position: 1 | a | a | Yes | Correctly extracts the first character. |
| Extract a middle character | + | Position: 17 | a | a | Yes | Correctly extracts a middle character. |
| Extract the last character | + | Position: 33 | a | ‘ ‘ | No | Fails to extract the last character; indicates an off-by-one error or improper handling of string length. |
| Attempt to extract beyond the string length | - | Position: 34 | Error or no output | ‘ ‘ | Yes | Correct behavior as there's no character at position 34. |
| Attempt to extract from a negative position | - | Position: -1 | Error or no output | a | No | Negative positions should not return valid characters; indicates incorrect handling of negative indices. |

**Test Cases for the White box program.**

| **Description** | **+ / − Purpose** | **Data Inputs for X and O** | **Expected Output** | | **Actual output if unexpected** | **Success?** | **Comments** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Start program | Record initial condition | n/a | 1 2 3  4 5 6  7 8 9 |  |  |  | To copy a grid from terminal, hold [Alt] while click & drag to select. |
| Nominal test | + check recording of alternating moves to open grid positions | X > 1 O > 2 | **X** 2 3  4 5 6  7 8 9 | X **O** 3  4 5 6  7 8 9 |  |  |  |
| Player O overwrites Player X's move | - | X > 1 O > 1 | Either an error message or prevention of the move |  | Grid shows O in position 1 after X's move was overwritten | No | The game should not allow overwriting of moves. This indicates a serious rule enforcement bug. |
| Normal gameplay with a winning move by Player X | + | X > 1 O > 2  X > 4 O > 5  X > 7 | X is declared the winner with a column win |  | X is declared the winner correctly | Yes | The game correctly identifies a win condition despite the earlier rule-breaking opportunity. |
| Player X completes a vertical line to win. | + | X > 1 O > 2  X > 3 O > 5  X > 4 O > 6  X > 7 | X wins with column 1-4-7. |  | X wins as expected. | Yes | The game correctly identifies and declares a winner based on a vertical column win. |
| Player O makes a move after Player X has already won. | - | O > 9 | No further moves allowed; game should be over. |  | Game allows O to take position 9 even after X has won. | No | The game should not allow moves after a win has been declared. This indicates a logic flaw that should be addressed. |