# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: 7

**1)Designing test cases for control structure statements.**

**MAIN ASPECT:**

**All possible inputs should be considered while testing**

**SPECIFICATIONS:**

* Test against a regular even number
* Test against a regular odd number
* Test against a zero
* Test against a negative even number
* Test against a negative odd number.
* Test against a large negative even number
* Test against a large negative odd number
* Test against a floating point number
* Test against a string

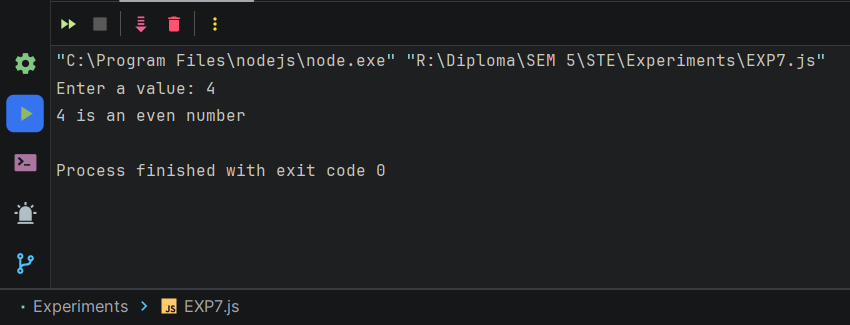
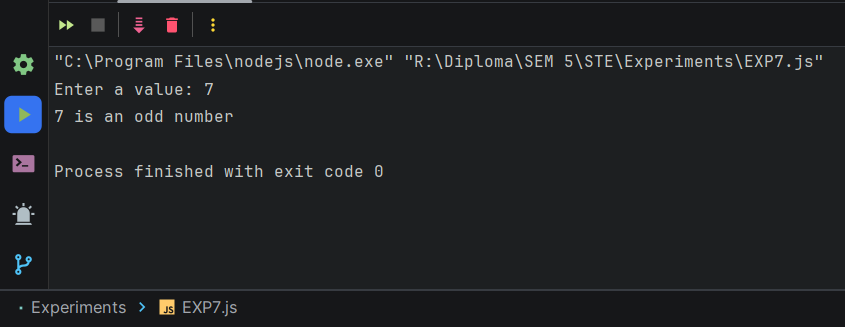
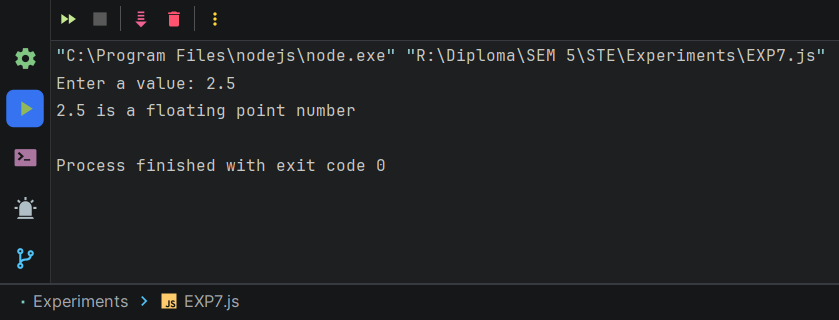
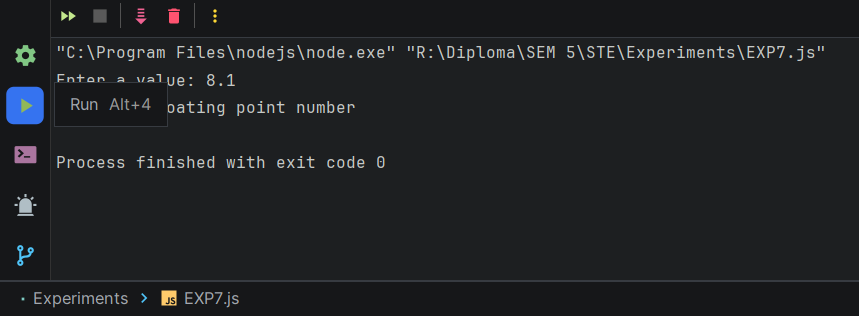
**REQUIREMENTS:**

* Program asks for an input from user
* All possible inputs are tested
* Corresponding result of corresponding value should be displayed

1. **CODE:**

const readline = ***require***('readline');  
  
const rl = readline.createInterface({  
 input: ***process***.stdin,  
 output: ***process***.stdout  
});  
  
rl.question('Enter a value: ', (userInput) => {  
 if (userInput === "") {  
 ***console***.log(`${userInput} is a string`);  
 } else {  
 const number = parseFloat(userInput);  
 if (!isNaN(number) && userInput.includes('.')) {  
 ***console***.log(`${userInput} is a floating point number`);  
 } else if (parseInt(userInput) % 2 === 0) {  
 ***console***.log(`${userInput} is an even number`);  
 } else if (parseInt(userInput) % 2 !== 0) {  
 ***console***.log(`${userInput} is an odd number`);  
 }  
 }  
 rl.close();  
});

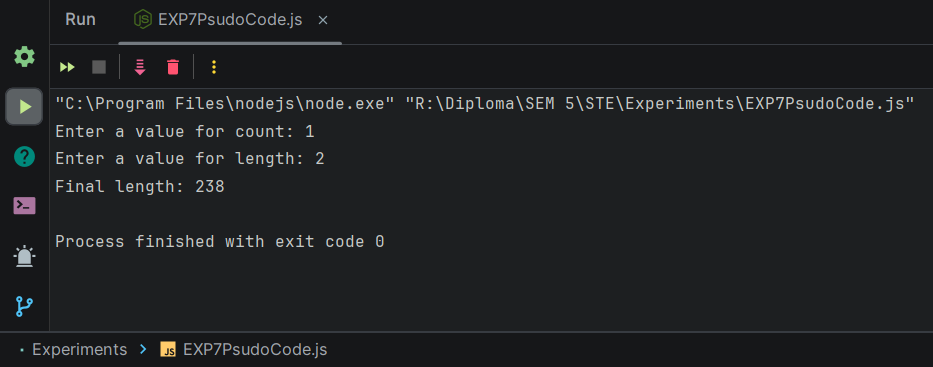
**OUTPUT:**

1. **CODE:**

const readline = ***require***('readline');  
  
let count;  
let length;  
  
const rl = readline.createInterface({  
 input: ***process***.stdin,  
 output: ***process***.stdout  
});  
  
rl.question('Enter a value for count: ', (userInputCount) => {  
 count = parseInt(userInputCount);  
  
 rl.question('Enter a value for length: ', (userInputLength) => {  
 length = parseInt(userInputLength);  
  
 rl.close();  
  
 while (count <= 6) {  
 if (length >= 100) {  
 length = length - 2;  
 } else {  
 length = count \* length;  
 }  
 count++;  
 }  
  
 ***console***.log(`Final length: ${length}`);  
 });  
});

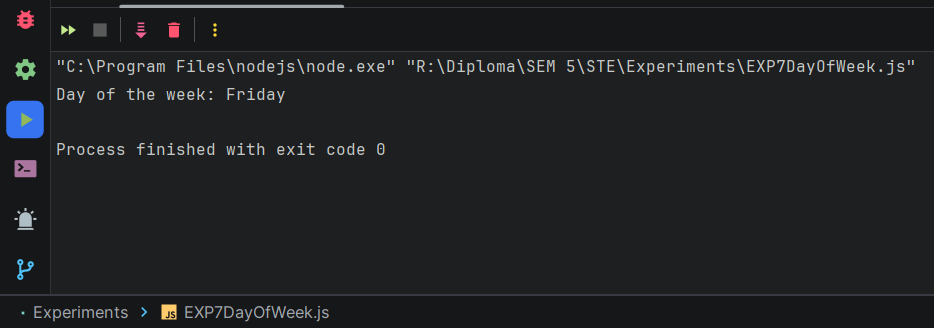
**OUTPUT:**



1. **CODE:**

const ***currentDate*** = new ***Date***();  
  
const ***daysOfWeek*** = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];  
  
const ***dayIndex*** = ***currentDate***.getDay();  
  
const ***dayName*** = ***daysOfWeek***[***dayIndex***];  
  
***console***.log("Day of the week: " + ***dayName***);

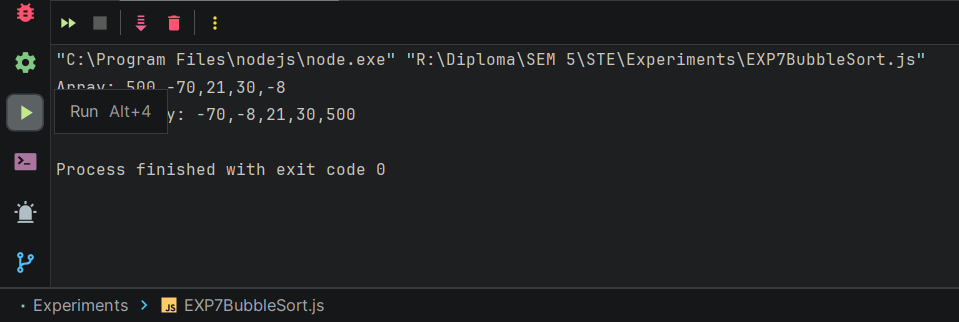
**OUTPUT:**



1. **CODE:**

const ***myArray*** = [500, -70, 21, 30, -8];  
***console***.log("Array: " + ***myArray***);  
const ***n*** = ***myArray***.length  
for (let i = 0; i < ***n*** - 1; i++) {  
 for (let j = 0; j < ***n*** - i - 1; j++) {  
 if (***myArray***[j] > ***myArray***[j + 1]) {  
 let temp = ***myArray***[j];  
 ***myArray***[j]=***myArray***[j+1];  
 ***myArray***[j+1]=temp;  
 }  
 }  
}  
***console***.log("Sorted Array: " + ***myArray***);

**OUTPUT:**



**CYCLOMATIC COMPLEXITY:**

E = 12

N = 10

P = 3

V(G) = E – N + 2

V(G) = 12 – 10 + 2 = 4