

Question 1

“V8 Engine” vs “Chakra” vs “Spider Monkey”

1) V8 Engine :-

Created by Google in 2008, written in C++, it is basically a JavaScript and WebAssembly engine that executes JavaScript codes, popularly known for powering Google Chrome.

Advantages:-

- 1- Efficient memory management, which is done by optimized Garbage collector.
- 2- High performance, uses JIT(Just-In-Time) compilation
- 3- Compatible with all platforms like Windows,Linux,macOS.

Disadvantages:-

- 1- High memory usage due to speed, uses more RAM.
- 2- Not ideal for low devices because it needs more memory
- 3- It is very complex internally and hard to debug.

2) Chakra :-

Created by Microsoft, was initially used for Internet Explorer as JScript, later on named as Chakra for Microsoft Edge. Currently, The ChakraCore support is inactive and community-driven.

Advantages:-

- 1- use JIT(Just-In-Time) compilation for faster execution speeds.
- 2- support parallel parsing and hence improves startup performance and responsiveness.

3- has good integration with vs code tools for debugging.

Disadvantages:-

- 1- Slower than V8 in many cases like performance tests
- 2- optimized only for windows, not for macOS and Linux
- 3- limited community and ecosystem, because development is discontinued.

3) SpiderMonkey :-

Developed by Brendan Eich in 1995, written in C++, SpiderMonkey was the first-ever JS engine, currently maintained by Mozilla and Now it powers the Firefox browser.

Advantages:-

- 1- It has advanced JIT compilation, because of its multiple Tiers JIT compilers.
- 2- Great debugging and developer tools.
- 3- It has a smart GC system, hence increases performance.

Disadvantages:-

- 1- not as fast as V8 in general use.
- 2- best optimized only for firefox
- 3- has less ecosystem support outside firefox

Summary :-

V8 is the winner here, because it has best performance and widespread adoption and it is available for all the platforms like Windows, MacOS, Linux. Also, most industries prefer it nowadays.

Question 2

HOF with 3 functions :-

```
function addAll() {
  let sum = 0;
  for (let i = 0; i < this.length; i++) {
    sum += this[i];
  }
  return sum; //function returns the sum of all elements
}

> function subtractAll() { ...
}

> function multiplyAll() { ...
}

function hof3(func1, func2, func3, numbers) { //first three functions are passed as arguments
  return {
    [func1.name]: func1.call(numbers),
    [func2.name]: func2.call(numbers),
    [func3.name]: func3.call(numbers),
  }; // hof3 takes three functions and an array of numbers
}

const result = hof3(addAll, subtractAll, multiplyAll, [1, 2, 3]);
console.log(result); // result will be displayed as an object with the results of each operation
```

HOF with 5 functions :-

```

function totalMarks() { // function to calculate total marks
  let sum = 0;
  for (let i = 0; i < this.length; i++) {
    sum += this[i];
  }
  return sum;
}

function averageMarks() { // function to calculate average marks
  let sum = 0;
  for (let i = 0; i < this.length; i++) {
    sum += this[i];
  }
  return (sum / this.length).toFixed(2);
}

function highestMarks() { // function to find the highest marks
  let highest = this[0];
  for (let i = 1; i < this.length; i++) {
    if (this[i] > highest) {
      highest = this[i];
    }
  }
  return highest;
}

function lowestMarks() { // function to find the lowest marks
  let lowest = this[0];
  for (let i = 1; i < this.length; i++) {

```

```

    lowest = this[i];
  }
}
return lowest;
}

> function passPercentage() { // function to know Pass or Fail...
}

// Higher-order function to analyze marks
function marksAnalyzerHOF(func1, func2, func3, func4, func5, marksArray) {
  return {
    [func1.name]: func1.call(marksArray),
    [func2.name]: func2.call(marksArray),
    [func3.name]: func3.call(marksArray),
    [func4.name]: func4.call(marksArray),
    [func5.name]: func5.call(marksArray),
  }; // hof takes five functions and an array of marks
}

const studentMarks = [95, 87, 84, 84, 91]; //sample data

const result = marksAnalyzerHOF(totalMarks, averageMarks, highestMarks, lowestMarks, passPercentage, studentMarks);

console.log(result);

```