**1. What will the output be for the following code?**

**function** test(callback1, callback2) {  
 callback1();  
 console.log("2");  
 callback1();  
 console.log("8");  
 callback2();  
 console.log("9");  
 callback2();  
}  
test(() **=>** console.log("3"), () **=>** console.log("6"));

1. 3 2 3 8 6 9 6
2. 3 3 2 6 8 6 9
3. 3 2 3 6 8 9 6
4. 3 3 6 2 8 9 6

**2. What will the following code output?**

**function** calculate(callback) {  
 **let** result = callback(5, 15, 25);  
 console.log(result);  
}  
calculate(**function** (x, y, z) {  
 **return** x \* y - z;  
});

1. 50
2. 25
3. 75
4. 100

**3. What is the value of result in this code?**

**function** sumValues(callback) {  
 **let** result = callback(2, 3, 4);  
 console.log(result);  
}  
sumValues((a, b, c) **=>** a + b \* c);

1. 14
2. 10
3. 9
4. 12

**4. What is printed to the console?**

**function** print(callback) {  
 callback();  
 console.log("Finished");  
}  
print(() **=>** console.log("Started"));

1. Started Finished
2. Finished Started
3. Error
4. Undefined

**5. What is the output of this function?**

**function** execute(callback1, callback2) {  
 callback1();  
 console.log("Middle");  
 callback2();  
}  
execute(() **=>** console.log("First"), () **=>** console.log("Second"));

1. First Middle Second
2. Middle First Second
3. First Second Middle
4. Error

**6. What will console.log output in this example?**

**function** operation(callback) {  
 **let** result = callback(8, 4);  
 console.log(result);  
}  
operation((a, b) **=>** a / b);

1. 12
2. 4
3. 2
4. 32

**7. What does the following code print?**

**function** runProcess(callback) {  
 callback();  
 console.log("Running...");  
}  
runProcess(() **=>** console.log("Started"));  
runProcess(() **=>** console.log("In Progress"));

1. Started Running… In Progress Running…
2. Started Running… Running… In Progress
3. Running… Started Running… In Progress
4. Started In Progress Running…

**8. What is the output for this function?**

**function** calc(callback) {  
 **let** result = callback(10, 5);  
 console.log(result);  
}  
calc(**function** (x, y) {  
 **return** x - y;  
});

1. 5
2. 15
3. 10
4. -5

**9. What will console.log print?**

**function** getResult(callback) {  
 **let** result = callback(6, 2);  
 console.log(result);  
}  
getResult((a, b) **=>** a % b);

1. 0
2. 1
3. 2
4. 3

**10. What is the value of result in this code?**

**function** compute(callback) {  
 **let** result = callback(7, 9, 3);  
 console.log(result);  
}  
compute((a, b, c) **=>** (a \* b) / c);

1. 21
2. 19
3. 27
4. 24

**11. What will the console output be?**

**function** process(callback) {  
 callback();  
 console.log("Processed");  
}  
process(() **=>** console.log("Initializing"));

1. Initializing Processed
2. Processed Initializing
3. Processed
4. Undefined

**12. What is the value of result printed in this code?**

**function** sum(callback) {  
 **let** result = callback(12, 8, 4);  
 console.log(result);  
}  
sum((x, y, z) **=>** x + y - z);

1. 16
2. 12
3. 24
4. 14

**13. What will the following code output?**

**function** multiply(callback) {  
 **let** result = callback(3, 5);  
 console.log(result);  
}  
multiply((x, y) **=>** x \* y);

1. 15
2. 8
3. 10
4. 5

**14. What is the value of result in this function?**

**function** subtract(callback) {  
 **let** result = callback(9, 4);  
 console.log(result);  
}  
subtract((a, b) **=>** a - b);

1. 13
2. 5
3. 6
4. 1

**15. What will the code print?**

**function** execute(callback1, callback2) {  
 callback1();  
 console.log("Step");  
 callback2();  
}  
execute(() **=>** console.log("Start"), () **=>** console.log("End"));

1. Start Step End
2. Step Start End
3. Start End Step
4. Error

**16. What does the following code print?**

**function** sumValues(callback) {  
 **let** result = callback(2, 4);  
 console.log(result);  
}  
sumValues((x, y) **=>** x + y);

1. 6
2. 8
3. 2
4. 4

**17. What is the result printed?**

**function** divide(callback) {  
 **let** result = callback(20, 4);  
 console.log(result);  
}  
divide((x, y) **=>** x / y);

1. 5
2. 4
3. 6
4. 16

**18. What will be the console output for this code?**

**function** showMessage(callback) {  
 callback();  
 console.log("Message shown");  
}  
showMessage(() **=>** console.log("Alert!"));

1. Alert! Message shown
2. Message shown Alert!
3. Alert!
4. Undefined

**19. What does the following code output?**

**function** calculateResult(callback) {  
 **let** result = callback(8, 2, 1);  
 console.log(result);  
}  
calculateResult((a, b, c) **=>** a - b + c);

1. 7
2. 5
3. 6
4. 9

**20. What is the value printed for result?**

**function** add(callback) {  
 **let** result = callback(5, 3);  
 console.log(result);  
}  
add((x, y) **=>** x + y);

1. 7
2. 8
3. 5
4. 2

**21. What will the output be for the following code?**

**function** funcA(callback1, callback2) {  
 callback1();  
 console.log("A1");  
 callback2();  
 console.log("A2");  
}  
  
**function** funcB(callback) {  
 console.log("B1");  
 callback();  
 console.log("B2");  
}  
  
funcA(() **=>** funcB(() **=>** console.log("Inside B")), () **=>** console.log("End of A"));

1. B1 Inside B B2 A1 End of A A2
2. A1 B1 Inside B B2 A2 End of A
3. A1 B1 Inside B End of A B2 A2
4. B1 A1 Inside B B2 A2 End of A

**22. What will be printed to the console by the following code?**

**function** outerFunc(callback) {  
 console.log("Outer Start");  
 callback();  
 console.log("Outer End");  
}  
  
**function** innerFunc(callback) {  
 console.log("Inner Start");  
 callback();  
 console.log("Inner End");  
}  
  
outerFunc(() **=>** innerFunc(() **=>** console.log("Innermost")));

1. Outer Start Inner Start Innermost Inner End Outer End
2. Inner Start Outer Start Innermost Inner End Outer End
3. Outer Start Innermost Inner Start Inner End Outer End
4. Outer Start Inner Start Inner End Outer End Innermost

**23. What is the result of the following code execution?**

**function** processA(callback1, callback2) {  
 callback1();  
 callback2();  
 console.log("ProcessA Done");  
}  
  
**function** processB(callback) {  
 console.log("Start ProcessB");  
 callback();  
 console.log("End ProcessB");  
}  
  
processA(() **=>** processB(() **=>** console.log("Inside ProcessB")), () **=>** console.log("End of ProcessA"));

1. Start ProcessB Inside ProcessB End ProcessB End of ProcessA ProcessA Done
2. Start ProcessB End ProcessB Inside ProcessB ProcessA Done End of ProcessA
3. Inside ProcessB Start ProcessB End ProcessB End of ProcessA ProcessA Done
4. Start ProcessB Inside ProcessB ProcessA Done End of ProcessA

**24. What will the following code print?**

**function** action1(callback1, callback2) {  
 callback1();  
 console.log("Action 1");  
 callback2();  
}  
  
**function** action2() {  
 console.log("Action 2");  
}  
  
**function** action3(callback) {  
 callback();  
 console.log("Action 3");  
}  
  
action1(() **=>** action3(() **=>** console.log("Start")), action2());

1. Start Action 3 Action 1 Action 2
2. Action 3 Start Action 2 Action 1
3. Start Action 3 Action 2 Action 1
4. Start Action 1 Action 3 Action 2

**25. What will be the final result of this code?**

**function** step1(callback) {  
 console.log("Step 1");  
 callback();  
}  
  
**function** step2(callback1, callback2) {  
 callback1();  
 console.log("Step 2");  
 callback2();  
}  
  
step2(() **=>** step1(() **=>** console.log("Inner Step 1")), () **=>** console.log("Inner Step 2"));

1. Step 1 Inner Step 1 Step 2 Inner Step 2
2. Inner Step 1 Step 1 Step 2 Inner Step 2
3. Step 1 Step 2 Inner Step 1 Inner Step 2
4. Step 1 Step 2 Inner Step 2 Inner Step 1

**26. What does this code print to the console?**

**function** first(callback) {  
 console.log("First");  
 callback();  
}  
  
**function** second(callback) {  
 console.log("Second");  
 callback();  
}  
  
**function** third() {  
 console.log("Third");  
}  
  
first(() **=>** second(third));

1. First Second Third
2. Second First Third
3. First Third Second
4. First Third Second Third

**27. What will the following code output?**

**function** alpha(callback1, callback2) {  
 console.log("Alpha Start");  
 callback1();  
 console.log("Alpha Middle");  
 callback2();  
 console.log("Alpha End");  
}  
  
**function** beta(callback) {  
 console.log("Beta Start");  
 callback();  
 console.log("Beta End");  
}  
  
**function** gamma() {  
 console.log("Gamma");  
}  
  
alpha(() **=>** beta(gamma), () **=>** console.log("Delta"));

1. Alpha Start Beta Start Gamma Beta End Alpha Middle Delta Alpha End
2. Alpha Start Gamma Beta End Alpha Middle Delta Alpha End
3. Alpha Start Gamma Delta Alpha Middle Beta End Alpha End
4. Alpha Start Beta Start Gamma Delta Alpha Middle Beta End Alpha End

**28. What is the output for this code?**

**function** run1(callback) {  
 console.log("Run1 Start");  
 callback();  
 console.log("Run1 End");  
}  
  
**function** run2(callback1, callback2) {  
 console.log("Run2 Start");  
 callback1();  
 console.log("Run2 Middle");  
 callback2();  
 console.log("Run2 End");  
}  
  
run2(() **=>** run1(() **=>** console.log("Run1 Inner")), () **=>** console.log("Run2 Inner"));

1. Run2 Start Run1 Start Run1 Inner Run1 End Run2 Middle Run2 Inner Run2 End
2. Run2 Start Run1 Start Run2 Inner Run1 End Run2 Middle Run2 End
3. Run2 Start Run2 Middle Run1 Start Run1 Inner Run1 End Run2 Inner Run2 End
4. Run2 Start Run1 Inner Run1 End Run2 Middle Run2 End

**29. What does the following code print?**

**function** firstAction(callback) {  
 console.log("Action 1");  
 callback();  
}  
  
**function** secondAction() {  
 console.log("Action 2");  
}  
  
**function** thirdAction(callback) {  
 console.log("Action 3");  
 callback();  
}  
  
firstAction(() **=>** thirdAction(secondAction));

1. Action 1 Action 3 Action 2
2. Action 3 Action 1 Action 2
3. Action 1 Action 2 Action 3
4. Action 2 Action 3 Action 1

**30. What is printed when the following code is executed?**

**function** start(callback1, callback2) {  
 callback1();  
 console.log("Start");  
 callback2();  
}  
  
**function** middle(callback) {  
 console.log("Middle");  
 callback();  
}  
  
**function** end() {  
 console.log("End");  
}  
  
start(() **=>** middle(() **=>** console.log("Beginning")), end());

1. Beginning Middle Start End
2. Middle Beginning Start End
3. Beginning Start Middle End
4. Middle Start Beginning End

**31. What will be the output of the following code?**

**function** funcA(callback1, callback2, callback3) {  
 console.log("A1");  
 callback1(() **=>** {  
 console.log("A2");  
 callback2();  
 console.log("A3");  
 });  
 console.log("A4");  
 callback3();  
}  
  
**function** funcB(callback) {  
 console.log("B1");  
 callback();  
 console.log("B2");  
}  
  
**function** funcC() {  
 console.log("C1");  
}  
  
funcA((innerCallback) **=>** funcB(innerCallback), () **=>** console.log("End of B"), funcC);

1. A1 B1 A2 End of B A3 B2 A4 C1
2. A1 B1 A2 B2 A3 End of B A4 C1
3. A1 A2 B1 B2 A3 A4 End of B C1
4. A1 A2 B1 B2 A4 A3 C1 End of B

**32. What will this program print to the console?**

**function** firstStep(callback1, callback2) {  
 console.log("First Step Start");  
 callback1(() **=>** {  
 console.log("First Step Mid");  
 callback2();  
 });  
 console.log("First Step End");  
}  
  
**function** secondStep(callback) {  
 console.log("Second Step Start");  
 callback();  
 console.log("Second Step End");  
}  
  
**function** thirdStep(callback) {  
 console.log("Third Step Start");  
 callback();  
 console.log("Third Step End");  
}  
  
firstStep(() **=>** secondStep(() **=>** console.log("Inside Second Step")), () **=>** thirdStep(() **=>** console.log("Inside Third Step")));

1. First Step Start Second Step Start Inside Second Step Second Step End First Step Mid Third Step Start Inside Third Step Third Step End First Step End
2. First Step Start Second Step Start Inside Second Step First Step Mid Third Step Start Inside Third Step Third Step End Second Step End First Step End
3. Second Step Start Inside Second Step Third Step Start Inside Third Step Third Step End First Step Start First Step End
4. First Step Start First Step End Second Step Start Inside Second Step Second Step End First Step Mid Third Step Start Inside Third Step Third Step End

**33. What will be printed by the following code?**

**function** alpha(callback1, callback2, callback3) {  
 console.log("Alpha Start");  
 callback1(() **=>** {  
 console.log("Alpha Mid");  
 callback2(() **=>** {  
 console.log("Alpha End");  
 callback3();  
 });  
 });  
}  
  
**function** beta(callback) {  
 console.log("Beta Start");  
 callback();  
 console.log("Beta End");  
}  
  
**function** gamma() {  
 console.log("Gamma Start");  
 console.log("Gamma End");  
}  
  
alpha((innerCallback) **=>** beta(innerCallback), (innerCallback) **=>** beta(innerCallback), gamma);

1. Alpha Start Beta Start Beta End Alpha Mid Beta Start Alpha End Gamma Start Gamma End
2. Alpha Start Beta Start Beta End Alpha Mid Alpha End Gamma Start Gamma End
3. Alpha Start Beta Start Alpha Mid Alpha End Beta End Gamma Start Gamma End
4. Alpha Start Alpha Mid Beta Start Beta End Alpha End Gamma Start Gamma End

**34. What is the result of the following code execution?**

**function** outer(callback1, callback2, callback3) {  
 console.log("Outer Start");  
 callback1(() **=>** {  
 console.log("Outer Mid");  
 callback2(() **=>** {  
 console.log("Outer End");  
 callback3();  
 });  
 });  
}  
  
**function** middle(callback) {  
 console.log("Middle Start");  
 callback();  
 console.log("Middle End");  
}  
  
**function** inner() {  
 console.log("Inner Action");  
}  
  
outer((cb) **=>** middle(cb), (cb) **=>** middle(cb), inner);

Ans: Outer Start, Middle Start, Outer Mid, Middle Start, Outer End ,Inner

Action, Middle End, Middle End

1. Outer Start Middle Start Middle End Outer Mid Middle Start Outer End Inner Action Middle End
2. Outer Start Middle Start Outer Mid Outer End Inner Action Middle Start Middle End
3. Outer Start Outer Mid Middle Start Middle End Outer End Inner Action
4. Outer Start Outer Mid Middle Start Outer End Inner Action Middle End

**35. What will be the final printed output?**

**function** main(callback1, callback2, callback3) {  
 console.log("Main Start");  
 callback1(() **=>** {  
 console.log("Main Mid");  
 callback2(() **=>** {  
 console.log("Main End");  
 callback3();  
 });  
 });  
}  
  
**function** sub1(callback) {  
 console.log("Sub1 Start");  
 callback();  
 console.log("Sub1 End");  
}  
  
**function** sub2() {  
 console.log("Sub2 Start");  
 console.log("Sub2 End");  
}  
  
main((cb) **=>** sub1(cb), (cb) **=>** sub1(cb), sub2);

Ans: Main Start, Sub1 Start, Main Mid, Sub1 Start, Main End, Sub2 Start, Sub2

End, Sub1 End, Sub1 End

1. Main Start Sub1 Start Sub1 End Main Mid Sub1 Start Main End Sub2 Start Sub2 End
2. Main Start Sub1 Start Sub1 End Main Mid Sub1 Start Sub2 Start Main End Sub2 End
3. Main Start Sub1 Start Main Mid Main End Sub1 Start Sub2 Start Sub1 End Sub2 End
4. Main Start Sub1 Start Sub1 End Main Mid Main End Sub2 Start Sub2 End