```
//CURRYING
// f(a,b) into f(a)(b)
function fx(a, b) {
  console.log(a, b);
}
function f(a) {
  return function (b) {
    return `${a} ${b}`;
  };
}
// console.log(f(5)(1));
// console.log(f(5));

function s(a, b, c) {
  return a + b + c;
}
function sum(a) {
  return function (b) {
    return function (c) {
      return a + b + c;
    };
  };
}
// console.log(sum(2)(6)(3));

function evaluate(operation) {
  return function (a) {
    return function (b) {
      if (operation === "sum") {
        return a + b;
      } else if (operation === "difference") {
        return a - b;
      } else if (operation === "multiply") {
        return a * b;
      } else if (operation === "divide") {
        return a / b;
      } else {
        return "Invalid Operation";
      }
    };
  };
}

// console.log(evaluate("sum")(109)(108));
const mul = evaluate("multiply");
// console.log(mul(17)(9));

//Infinite Currying
```

```
function addinfi(a) {
  return function (b) {
    if (b) {
      return addinfi(a + b);
    } else return a;
  };
}
// console.log(addinfi(1)(2)(3)(4)());
//Currying vs Partial Application
function fy(a) {
  return function (b, c) {
    return a + b + c;
  };
}
// console.log(fy(2)(3, 5));
//This is partial application of currying

//Practical application of currying
function updateElemText(id) {
  return function (content) {
    document.querySelector(`#${id}`).textContent = content;
  };
}
const updateHeaderText = updateElemText("header");
// updateHeaderText("Hello RoadsideCoder!");

//Curry from f(a,b,c) to f(a)(b)(c)

function curry(func) {
  return function curriedFunc(...args) {
    if (func.length <= args.length) {
      return func(...args);
    } else {
      return function (...next) {
        return curriedFunc(...args, ...next);
      };
    }
  };
}
function sums(a, b, c, d) {
  return a + b + c + d;
}
const totalSum = curry(sums);
// console.log(totalSum)
console.log(totalSum(1)(3)(5)(7));
```