// //const url = '<https://us-central1-exploration-pocs.cloudfunctions.net/candidate-interview/api/uids>';

// const data=fetch('<https://us-central1-exploration-pocs.cloudfunctions.net/candidate-interview/api/uids>')

//   .then(response => response.json())

//   .then(data => {

//     console.log(data);

//   })

//   .catch(error => console.error(error));

//ReferenceError: fetch is not defined

const data = '{"isSuccess":true,"uids":{"pallet-1":{"case-1":{"bundle-1":{"bottle-1":null,"bottle-2":null},"bundle-2":{"bottle-3":null,"bottle-5":null}},"case-2":{"bundle-3":{"bottle-6":null},"bundle-4":{"bottle-7":null,"bottle-8":null}}},"case-3":{"bundle-5":{"bottle-9":null},"bundle-6":{"bottle-10":null}},"pallet-3":{"case-5":{"bundle-7":{"bottle-11":null},"bundle-8":{"bottle-12":null}}}}}';

const childId = "bottle-1";

let parentId = null;

const dataObj = JSON.parse(data);

function getParentId(childId) {

  //console.log(dataObj)

  for (let outer1 in dataObj.uids) {

    const res1 = dataObj.uids[outer1];

    //console.log(res1)

     if (res1 && res1.hasOwnProperty(childId)) {

      parentId = outer1;

      break;

    }

    for (let outer2 in res1) {

      const res2 = res1[outer2];

   // console.log(res2)

      if (res2 && res2.hasOwnProperty(childId)) {

        parentId = outer2;

        break;

      }

      for (let outer3 in res2) {

        const res3 = res2[outer3];

        //console.log(res3)

        if (res3 && res3.hasOwnProperty(childId)) {

          parentId = outer3;

          break;

        }

      }

       if (parentId) {

      break;

    }

    }

    if (parentId) {

      break;

    }

  }

}

getParentId(childId);

console.log(parentId); //output's the immediate parent

//  console.log('Test 1 passing:');

//   console.log(await getParentId("bottle-1") === "bundle-1");

//   console.log('Test 2 passing: ');

//   console.log(await getParentId("bundle-2") === "case-1");

//   console.log('Test 3 passing: ');

//   console.log(await getParentId("case-2") === "pallet-1");

//   console.log('Test 4 passing: ');

//   console.log(await getParentId("pallet-2") === null);

//   console.log('Test 5 passing: ');

//   console.log(await getParentId("case-5") === "pallet-3");

//   console.log('Test 6 passing: ');

//   console.log(await getParentId("bottle-12") === "bundle-8")

;

=======================================================================

const arr = [1,4,3];

let res = '';

function OtherProducts(arr){

for (let i = 0; i < arr.length; i++) {

  let product = 1;

  for (let j = 0; j < arr.length; j++) {

    if (i !== j) {

      product \*= arr[j];

    }

  }

  res += product;

  if (i !== arr.length - 1) {

    res += '-';

  }

}

console.log(res)

}

OtherProducts(arr)

===============================================================

//var str="neveroddoreven"

function Palindrome(str){

console.log(str)

   var res = str.split("").reverse().join("");

   if(str==res){

    //  console.log("Palindrome")

    //  p =="Palindrome";

    return true;

   }else{

// console.log("Not a Palindrome")

return false;

   }

  //return false;

}

//palindrome(str);

// keep this function call here

 console.log('Test 1 passing: ' + (Palindrome("neveroddoreven") === true));

 console.log('Test 2 passing: ' + (Palindrome("eye") === true));

 console.log('Test 3 passing: ' + (Palindrome("not\_a\_palindrome") === false));