2. Write a program to find out whether a given number is a square.

<!DOCTYPE html>

<html>

<head>

<title>Question 1.2</title>

<script type="text/javascript">

function checkSquare(){

var num = document.getElementById("num").value;

var square = false;

if(num == 1){

square = true;

}

for(var i = 1; i < num; i++){

if(i\*i == num){

square = true;

i = num;

}

}

if (square == true){

document.write(num + " is a square");

}

else{

document.write(num + " is not a square");

}

}

</script>

<meta charset="utf-8">

</head>

<body>

<h1>Testing for a square</h1><br/>

<h2>What number would you like to test?</h2><br/>

<form name="numForm">

<input type="text" id="num"/>

</form>

<button onclick="checkSquare()">Check</button>

</body>

</html>

4 a. Verify the identity .











b. Write the fraction as (a times) and then use the identity of part (a) repeatedly until all of the fractions are distinct.

<!DOCTYPE HTML>

<html>

<head>

<meta charset="UTF-8">

<title>Representing fractions </title>

<script>

var denoms = []; //will hold the denominators

function expand(){

var a = parseInt(document.f.an.value);

var b = parseInt(document.f.bn.value);

var j;

//alert("a is "+a+" and b is "+b);

//build up denoms

for (var i=0;i<a;i++){

denoms.push(b);

} //end initial for loop

//start to work on denoms, expanding any repeats

j = 1;

while (j<denoms.length){

//check if jth element is a repeat of any prior elements

jok = true;

for (var k=0;k<j;k++){

denomAtJ = denoms[j];

if (denomAtJ==denoms[k]) {

//need to change denom j

denoms.splice(j,1,denomAtJ+1,denomAtJ\*(denomAtJ+1));

jok = false;

break;

} //ends if

}

if (jok){j++;}

} //ends while

//print out answer using denoms

answerst ="";

for (var i=0;i<denoms.length;i++){

answerst+="<br/> 1 / "+denoms[i];

}

document.getElementById("answer").innerHTML = answerst;

denoms = []; //prepare for next task

return false;

} //ends function

</script>

</head>

<body>

<form name="f" onsubmit="return expand();">

Enter numerator and denominator (positive integers, please) <br/>

<input type="number" name="an" min="1"/> <br/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<br/>

<input type="number" name="bn" min="1"/>

<br/>

<input type="submit" value="Submit"/>

</form>

<br/>

Expansion of fraction is: <br/>

<div id="answer">

</div>

</body>

</html>

6.) Find 100 triangular numbers that are squares.

\*\*Due to the limited amount of square triangular numbers, code might not complete on all computers

<!DOCTYPE html>

<html>

<head>

<title>Question 1.6</title>

<script type="text/javascript">

function triangleNumbers(){

var number = 1;

var count = 0;

var check = 0;

while(count < 100){

var varify = false;

while(!varify){

check = (number\*(number + 1))/2;

console.log(check);

console.log("Is square is " + isSquare(check));

if(isSquare(check)){

varify = true;

}

console.log(number + " " + count);

number++;

}

count++;

document.write(count + ".) " + check + "<br/>");

}

}

function isSquare(num){

var n = Math.sqrt(num);

if(n%1 == 0){

return true;

}

else{

return false;

}

}

</script>

<meta charset="utf-8">

</head>

<body onload="triangleNumbers()">

<h1>Square Triangle Numbers</h1><br/>

<h2>Click to find the first 100 triangle numbers</h2><br/>

</body>

</html>

8.) Find 100 oblong numbers which are products of an oblong number and a square.

<!DOCTYPE HTML>

<html>

<head>

<title>Finding certain Oblong Numbers </title>

<script>

var oblongs = Array();

oblongs.push(2);

oblongs.push(6);

function genOblongAndCheck(){

found = 0;

while (found<100){

//add next oblong number

n = oblongs.length;

nextoblong = (n+1)\*n;

//see if nextoblong is divisible by any oblong in oblongs

for(var j=0;j<n;j++){

quotient = nextoblong / oblongs[j];

if (isInteger(quotient)){

//nextoblong is divisible by a previous oblong.

//now check if the quotient is a square

if (ans=isSquare(quotient)){

found++;

document.write(String(nextoblong)+" is the oblong "+String(n+1)+" times "+String(n)+" and is equal to the "+String(j+1)+"th oblong times "+ String(ans)+ " squared.<br/>" );

break;

}

}

}

oblongs.push(nextoblong);

}

}

function isInteger(n){

return n % 1 === 0;

}

function isSquare(n){

var r = Math.sqrt(n);

if (isInteger(r)){

return (r); //non-zero number interpreted as true

}

else {

return false;

}

}

</script>

<body onload="genOblongAndCheck();">

Determining which oblongs are product of oblong and a square

</body>

</html>

10.) Write each of the numbers from 1 to 1000 as the sum of three or fewer triangular numbers.

<!DOCTYPE html>

<html>

<meta charset="utf-8">

<head>

<title>Question 1.10</title>

<script type="text/javascript">

function sumNumbers(){

var triNumbers = [];

for(j = 1; j <= 1000; j++){

var pos = triNumbers.length - 1;

var sums = [];

//create the sums

while(j > triNumbers[triNumbers.length - 1] || triNumbers.length == 0){

triNumbers = addTriangle(triNumbers);

pos++;

}

var currentNum = triNumbers[pos];

var dif;

var second = true;

if(currentNum == j){

sums.push(currentNum);

second = false;

}

else{

var flag = true;

while(flag){//looking for 2 numbers to add together

if(pos < 0){

flag = false;

}

currentNum = triNumbers[pos];

dif = j - currentNum;

if(dif > 0 && dif <= currentNum){

var check = true;

var i = 0;

while(check){

if(triNumbers[i] == dif){

sums.push(currentNum);

sums.push(triNumbers[i]);

check = false;

flag = false;

second = false;

}

else if(triNumbers[i] > dif){

check = false;

pos--;

}

else{

i++;

}

}

}

else if(dif > currentNum){//exit this loop

dif = j - triNumbers[pos +1];

sums.push(triNumbers[pos + 1]);

flag = false;

}

else if(dif == 0){//end process, solution found

flag = false;

second = false;

sums.push(currentNum);

}

else{//dif negative

pos--;

}

}

}

if(second){

var count = 0;

var firstDif = dif;

var flag = true;

pos = triNumbers.length - 1;

while(flag){

currentNum = triNumbers[pos];

dif = firstDif - currentNum;

var check = true;

var i = 0;

while(check){

if(triNumbers[i] == dif){

sums.push(currentNum);

sums.push(triNumbers[i]);

check = false;

flag = false;

}

else if(triNumbers[i] > dif){

pos--;

check = false;

}

else{

i++;

count++;

if(count > 500){

check = false;

flag = false;

}

}

}

}

}

document.write(printAdds(sums, j) + "<br/>");

}

}

function addTriangle(array){

if(array.length == 0){

array.push(1);

}

else{

var toAdd;

toAdd = array[array.length - 1] + (array.length + 1);

array.push(toAdd);

}

return array;

}

function printAdds(nums, sol){

var addsString = "";

for(var i = 0; i < nums.length; i++){

if(i < nums.length - 1){

addsString = addsString + nums[i] + " + ";

}

else{

addsString = addsString + nums[i];

}

}

addsString = addsString + " = " + sol;

return addsString;

}

</script>

</head>

<body onload="sumNumbers()">

<h1>Numbers 1 through 1000 is triangle number sums</h1><br/>

</body>

</html>

12.) What proportion of the first one million numbers can be written using two or fewer squares?

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.12</title>

<script type="text/javascript">

function squareSum(){

var squares = [];

var numerator = 0;

var denominator = 1000000;

for(var i = 1; i < 1000001; i++){

var check = false;

var squPos = squares.length - 1;

var currentNum;

var numberOfNums = 0;

var sum = 0;

//var nums = [];

while(!check){

while(i > squares[squares.length - 1] || squares.length == 0){

squares = addSquare(squares);

squPos++;

}

currentNum = squares[squPos];

if(currentNum + sum < i){

sum = sum + currentNum;

numberOfNums++;

//nums.push(currentNum);

}

else if(currentNum + sum > i){

squPos--;

}

else{//current num == sum

sum = sum + currentNum;

numberOfNums++;

//nums.push(currentNum);

check = true;

if(numberOfNums <= 2){

numerator++;

//printAddition(nums, i);

}

}

}

}

document.write("The ratio is \n" + numerator + "/" + denominator);

}

function addSquare(array){

var square;

var last = array.length + 1;//the next oblong number to be added (skip 1)

square = last\*last;

array.push(square);

return array;

}

</script>

</head>

<body onload="squareSum()">

<h1>Proportion for 2 or fewer squares</h1><br/>

</body>

</html>

14. Write a program to find the sum of the proper divisors of a given number.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.14</title>

<script type="text/javascript">

function sumDivisors(){

var num = document.getElementById("num").value;

var sum = 0;

var div = num - 1;

var divisors = [];

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

document.write("The sum of the proper divisors of " + num + " is " + sum);

}

</script>

</head>

<body>

<h1>Sum of Proper Divisors</h1><br/>

<h2>Please enter a number</h2><br/>

<form name="numForm">

<input type="text" id="num"/>

</form>

<button onclick="sumDivisors()">Check</button>

</body>

</html>

16. Find the next three perfect numbers after 8128.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.16</title>

<script type="text/javascript">

function findPerfects(){

var num = 33550330;

var trueCount = 0;

var perfects = []

while(trueCount < 1){

num++;

var div = num - 1;

var divisors = [];

//find the divisors

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

var sum = 0;

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

if(sum == num){

trueCount++;

perfects.push(num);

}

}

document.write("The next perfect number is " + perfects);

}

</script>

</head>

<body onload="findPerfects()">

<h1>Finding Perfect Numbers after 8128</h1><br/>

</body>

</html>

18. If a given number is abundant, determine if it is semi-perfect.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.18</title>

<script type="text/javascript">

function semiPerfect(){

var num = document.getElementById("num").value;

//Test for abundence

var div = num - 1;

var divisors = [];

var sum = 0;

var abund = false;

var semi = false;

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

if(sum > num){

abund = true;

sum = 0;//reset sum

var sums = [];

var divPos = 0;

var currentNum = divisors[divPos];

while(divPos >= 0 && semi == false){

if(sum + currentNum < num){

sum = sum + currentNum;

sums.push(currentNum);

divPos++;

}

else if(sum + currentNum == num){

semi = true;

sums.push(currentNum);

sum = sum + currentNum;

}

else{//sums + currentNum > num

divPos++;

}

currentNum = divisors[divPos];

}

}

else{

document.write("The number is not abundent");

}

if(abund){

if(semi){

document.write("The number is semiPerfect<br/>The divisors are " + printAddition(sums, num));

}

else{

document.write("The number is not semiPerfect");

}

}

}

function printAddition(nums, total){

var addString = "";

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

if(i > 0){

addString = addString + " + " + nums[i];

}

else{

addString = "" + nums[i];

}

}

addString = addString + " = " + total;

return addString;

}

</script>

<meta charset="utf-8">

</head>

<body>

<h1>Semi-Perfect Abundant Numbers</h1><br/>

<h2>Please enter a number</h2><br/>

<form name="numForm">

<input type="text" id="num"/>

</form>

<button onclick="semiPerfect()">Check</button>

</body>

</html>

20. Verify that 945 is the smallest odd abundant number. Find the next 50 odd abundant numbers. Do they seem to be getting further apart?

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.20</title>

<script type="text/javascript">

function abundentOdd(){

//varify 945 is smallest odd

var num = 943

var varify = true;

//Varify Abundent Odds

while(num > 0 && varify){

var div = num - 1;

var divisors = [];

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

var sum = 0;

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

if(sum > num){

document.write("The abundent odd is " + num);

varify = false;

}

num = num - 2;

}

if(varify){

document.write("Has been varified");

}

//Find next 100 Abundent Odds

num = 947

var numCount = 0;

var abunds = [];

while(numCount < 100){

var div = num - 1;

var divisors = [];

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

var sum = 0;

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

if(sum > num){

abunds.push(num);

numCount++;

}

num = num + 2;

}

for(var i = 0; i < abunds.length; i++){

document.write("<br/>" + (i+1) + ".) " + abunds[i]);

}

}

</script>

</head>

<body onload="abundentOdd()">

<h1>Abundent and Odd</h1>

</body>

</html>

22. It has been conjectured that no odd number is perfect. Verify this for the first 1000 odd numbers. *See last example. You can decide on other ways to present the findings and also change the limit.*

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Question 1.22</title>

<script type="text/javascript">

function perfectOdds(){

var varified = true;

var count = 0;

var num = 1;

while(count < 1000 && varified){

var div = num - 1;

var divisors = [];

while(div > 0){

if(num%div == 0){

divisors.push(div);

}

div--;

}

var sum = 0;

for(var i = 0; i < divisors.length; i++){

sum = sum + divisors[i];

}

if(sum == num){

varified = false;

console.log("The number " + num + " is perfect");

}

count++;

num = num + 2;

}

if(varified){

document.write("None of the first 1000 odds are perfect");

}

else{

document.write("The number " + num + " is perfect");

}

}

</script>

</head>

<body onload="perfectOdds()">

<h1>Perfect Odds?</h1><br/>

</body>

</html>