1. ***Factorial program using typescript***

function fact(n: number): number {

let f1: number = 1;

for (let i: number = n; i >= 1; i--) {

f1 = f1 \* i;

}

return f1;

}

let n: number = 5;

let f1: number = fact(n);

console.log("Factorial is: " + f1);

***output:***

Factorial is: 120

***2.find the area circle and area of sphere use interface.***

interface Shape {

r: number;

}

interface Circle extends Shape {

h: number;

}

let circle: Circle = {

r: 5,

h: 10,

};

console.log(

`radius is : ${circle.r} . the area of circle is : ${

3.14 \* circle.r \* circle.r

}`

);

console.log(` the area of sphere is : ${4 \* 3.14 \* circle.r \* circle.r}`);

***output:***

radius is : 5 . the area of circle is : 78.5

the area of sphere is : 314

***3.check Armstrong using typescript***

function armstrong(n: number): string {

let p: number;

let sum: number = 0;

let n1: number;

p = n;

while (n > 0) {

n1 = n % 10;

n = Math.floor(n / 10);

sum = sum + n1 \* n1 \* n1;

}

if (sum == p) {

return "number is armstrong";

} else {

return "number is not armstrong";

}

}

{

let n: number = 135;

console.log(" The number is " + armstrong(n));

}

***Output:***

The number is number is not Armstrong

***4.Area of circle use typescript***

function circle(r:number):number {

let area:number;

area = 3.14 \* r \* r;

return area;

}

let r:number = 4;

console.log("area is: " + circle(r));

***output:***

area is: 50.24

***5.check number is pronic or not***

function pronic\_r(num:number):string {

let isPronic:boolean = false;

for (let i:number = 1; i <= num - 1; i++) {

if (i \* (i + 1) == num) {

isPronic = true;

break;

}

}

if (isPronic) return num + " is a pronic number";

else return num + " is not a pronic number";

}

let num:number = 20;

console.log(" " + pronic\_r(num));

***output:***

20 is a pronic number

***6.find the area and volume of cylinder using type alice***

type size = {

radius: number;

height: number;

};

let cylinder: size = {

radius: 10,

height: 20,

};

console.log(

The radius of cylinder is ${cylinder.radius} and height is ${cylinder.height}

);

console.log(

` the area of cylinder is ${

2 \* 3.14 \* cylinder.radius \* (cylinder.radius + cylinder.height)

}`

);

console.log(

` the volume is cylinder is ${

3.14 \* cylinder.radius \* cylinder.radius \* cylinder.height

}`

);

***Output:***

The radius of cylinder is 10 and height is 20

the area of cylinder is 1884.0000000000002

the volume is cylinder is 6280

***7.check number is palindrome or not use typescript***

function pal(n:number):string {

let p:number = n;

let sum:number = 0;

while (p > 0) {

let n1: number = p % 10;

p = Math.floor(p / 10);

sum = sum \* 10 + n1;

}

if (sum == n) return "Number is pal";

else return "Number is not pal";

}

let n:number = 121;

console.log(" " + pal(n));

***output:***

Number is pal

***8.find power use typescript***

function power(x:number, n:number):number {

let f1:number = 1;

for (let i:number = 1; i <= n; i++) {

f1 = f1 \* x;

}

return f1;

}

let n:number =2;

let x:number =4;

console.log("number" + n + " power " + x + " is = ", power(x, n));

***output:***

number2 power 4 is = 16

***9.find gcd and lcm***

function gcd(x:number, y:number) {

let gcd:number = 1;

for (let i:number = 2; i <= x && i <= y; i++) {

if (x % i == 0 && y % i == 0) {

gcd = i;

}

}

return gcd;

}

let x:number = Number(prompt("enetr 1 number:"));

let y:number = Number(prompt("enetr 2 number:"));

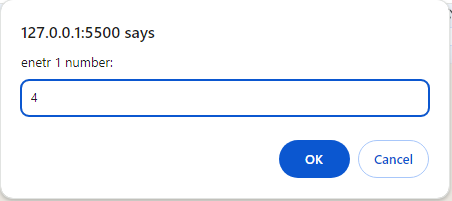
let GCD = gcd(x, y);

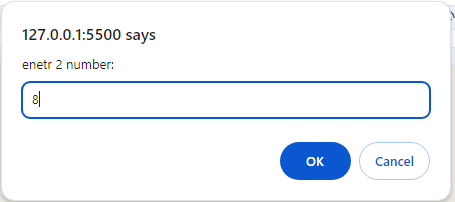
console.log("GCD is:", +GCD);

let lcm = (x \* y) / GCD;

console.log("LCM IS :", lcm)

***output:***





GCD is: 4

LCM IS : 8

***10. find arithmetic mean and harmonic mean***

function mean(a: number, b: number) {

let arithMean: number = (a + b) / 2;

let harmonic: number = (a - b) / 2;

return { arithMean, harmonic };

}

let a: number = Number(prompt("enter a"));

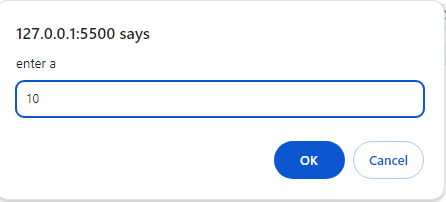
let b: number = Number(prompt("enter b"));

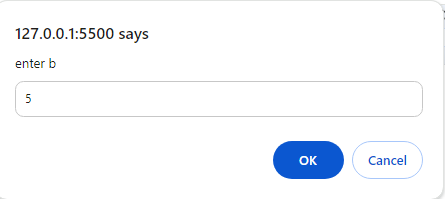
let result = mean(Number(a), Number(b));

console.log("arithmatic mean :" + result.arithMean);

console.log("harmonic mean is" + result.harmonic);

***Output:***





arithmatic mean :7.5

harmonic mean is2.5

***11.find the total and average of marks***

function marks\_total(m1:number, m2:number, m3:number) {

let total:number = m1 + m2 + m3;

let avg:number = total / 3;

return { total, avg };

}

let m1 = prompt("enter m1");

let m2 = prompt("enter m2");

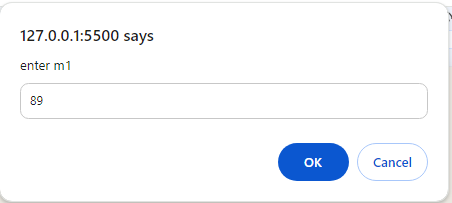
let m3 = prompt("enter m3");

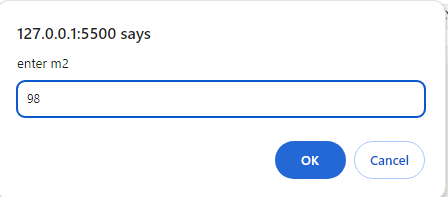
let result = marks\_total(Number(m1), Number(m2), Number(m3));

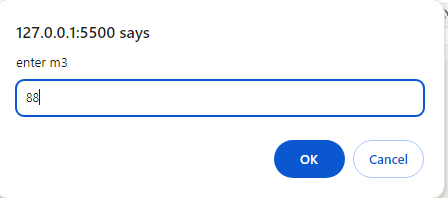
console.log("total is:" + result.total);

console.log("average is:" + result.avg);

***output:***







total is:275

average is:91.66666666666667

***12.find potential energy use typescript***

function potential(m:number, g:number, h:number) {

let PE:number = m \* g \* h;

return PE;

}

let m:number = 9;

let g:number = 6;

let h:number = 12;

console.log("potential energy is :" + potential(m, g, h));

***output:***

potential energy is :648

***13. Display the book information***

interface Book {

bookId: number;

bookName: string;

bookAuthor: string;

bookPrice: number;

}

interface book extends Book{

bookPublisher: string;

}

let book1: book = {

bookId: 1,

bookName: "ramayan",

bookAuthor: "valmiki",

bookPrice: 1000,

bookPublisher: "xyz"

};

console.log(book1.bookPublisher);

console.log(book1.bookId);

console.log(book1.bookName);

console.log(book1.bookAuthor);

console.log(book1.bookPrice);

output:

xyz

1

ramayan

valmiki

1000

***14.find hra da***

let salary = (bs:number) => {

let hra:number = bs \* 0.1;

let da :number= bs \* 0.9;

return { hra, da };

};

let bs = 20000;

console.log("hra is :" + salary(bs).hra);

console.log("da is :" + salary(bs).da);

***output:***

hra is :2000

da is :18000

***15.student information using interface use typescript***

interface Student {

id: number;

name: string;

age: number;

}

interface stud extends Student {

grade: string;

courses: string[];

}

const student1: stud = {

id: 23,

name: "isha",

age: 23,

grade: "A+",

courses: ["bca", "bcs", "bsc"],

};

console.log("Student ID:", student1.id);

console.log("Name:", student1.name);

console.log("Age:", student1.age);

console.log("Grade:", student1.grade);

console.log("Courses:", student1.courses);

***output:***

Student ID: 23

10 Name: isha

11 Age: 23

Grade: A+

Courses: (3) ['bca', 'bcs', 'bsc']

***16.display employee information use typescript***

type emp={

id:number,

name:string,

salary:number

dateOfJoining:number

}

let emp1:emp = {

id: 1,

name: "isha",

salary: 20000,

dateOfJoining: 2020

}

console.log("Employee ID:", emp1.id);

console.log("Employee Name:", emp1.name);

console.log("Employee Salary:", emp1.salary);

console.log("Employee Date of Joining:", emp1.dateOfJoining);

***output:***

Employee ID: 1

Employee Name: isha

10 Employee Salary: 20000

Employee Date of Joining: 2020

**17.prime find using typescript**

function prime(n:number):string {

let flag:number = 0;

for (let i:number = 2; i <= n / 2; i++) {

if (n % i == 0) {

flag = 1;

break;

}

}

if (flag == 0) {

return "number is prime";

} else {

return "number is not prime";

}

}

{

let n:number = 45;

console.log(" " + prime(n));

}

***Output:***

number is not prime

***18.sum of digit use typescript***

function sum\_pr(n: number): number {

let n1: number,

sum: number = 0;

while (n > 0) {

n1 = n % 10;

sum += n1;

n = Math.floor(n / 10);

}

return sum;

}

let n: number = 110;

console.log("Sum :" + sum\_pr(n));

***output:***

Sum : 2

***19. Write a program to using marker interface product(product\_id, product\_name, product\_cost, product\_quantity)***

interface IntOperations {

num: number;

isPositive(): boolean;

isNegative(): boolean;

isEven(): boolean;

isOdd(): boolean;

factorial(): number;

sumOfDigits(): number;

}

let num: IntOperations = {

num: 5,

isPositive: function(): boolean {

return this.num > 0;

},

isNegative: function(): boolean {

return this.num < 0;

},

isEven: function(): boolean {

return this.num % 2 == 0;

},

isOdd: function(): boolean {

return this.num % 2 != 0;

},

factorial: function(): number {

let result = 1;

for (let i = 1; i <= this.num; i++) {

result \*= i;

}

return result;

},

sumOfDigits: function(): number {

let sum = 0;

let temp = this.num;

while (temp > 0) {

sum += temp % 10;

temp = Math.floor(temp / 10);

}

return sum;

}

};

console.log(num.isPositive());

console.log(num.isNegative());

console.log(num.isEven());

console.log(num.isOdd());

console.log(num.factorial());

console.log(num.sumOfDigits());

***output:***

true

false

false

true

120

5

***20.find cube***

let cube = (a:number) => a \* a \* a;

let a = 25

console.log("cube is :" + cube(a));

***output:***

cube is :15625

***21.display information use type alices***

type Person ={

name: string;

age: number;

employeeId: number;

salary: number;

}

let emp: Person = {

name: "Rahul",

age: 25,

employeeId: 101,

salary: 5000

};

console.log(emp.employeeId);

console.log(emp.name);

***output:***

101

Rahul