function someEg1(a:(number)=>string, ...data:number[])

{

let x = 0;

while(x<data.length){

console.log(a(data[x]));

x++;

}

}

function eg1\_1(x:number)

{

return `cool ${x}`;

}

function eg1()

{

someEg1(eg1\_1, 10,20,30,40);

}

eg1();

In the above code a:(number)->string we can change with our own custom type

type myTypeinEg2=(n:number)=>string;

function someEg2(a:myTypeinEg2, ...data:number[])

{

let x = 0;

while(x<data.length)

{

console.log(a(data[x]));

x++;

}

}

function eg2\_1(a:number){

return `cool ${a}`;

}

function eg2(){

someEg2(eg2\_1, 20, 20,30 ,40);

}

eg2();

If the function type is void then you can not return anything

function someEg3():void

{

console.log(100);

return 50; // incorrect

}

If the function return type is never:

//The following someEg4 is incorrect because it has a reachable endpoint

function someEg4():never

{

console.log("Great");

}

If you have given type never, then this function should not have an endpoint

Final code:

function someEg5():never

{

console.log("Great");

throw new Error ("whatever");

}

**ARRAY**

Let x:number[];

Let x:(number | string)[]

Let x: number[] | string[]

Let x:number[][]; //2d array

function eg8(){

let x:(number | string)[];

x = [10,20,30];

console.log(x);

let y:(number | string)[];

y = ["God","Is","Great"];

console.log(y);

let z:(number | string)[];

z = ["God is great", 10];

console.log(z);

z = [20, "God is great", 10];

console.log(z);

let p: number[] | string[];// p can be either an array of string or number not mix

p = [10, "God is great"]; // incorrect

console.log(p);

p = [ "God is great", 10]; // incorrect

console.log(p);

p = [ "God is great", "Ujjain is city"]; // correct

console.log(p);

p = [ 100, 200, 10]; // correct

console.log(p);

}

eg8();

/\*\*

\* eg9.ts:15:5 - error TS2322: Type '(string | number)[]' is not assignable to type 'string[] | number[]'.

Type '(string | number)[]' is not assignable to type 'string[]'.

Type 'string | number' is not assignable to type 'string'.

Type 'number' is not assignable to type 'string'.

15 p = [10, "God is great"];

~

Found 1 error in eg9.ts:15

\*/

function eg10(){

let x:number |string[];

x = 10;

console.log(x);

x = ["God", "is",'great'];

console.log(x);

x = [10, "god"]; // incorrect

console.log(x);

x = ["GOd", 10]; // incorrect

console.log(x);

}

eg10();

function eg11(){

let x = [

[10,20,30],

["god","is","great"],

[true, 20]

];

for (let a = 0; a < x.length; a++) {

for (let b = 0; b < x[a].length; b++) {

console.log(x[a][b]);

}

}

}

eg11();

function eg12(){

let x:number[][];

x = [

[10,20],

[30,40,50],

[60,70]

]

console.log(x);

}

eg12();

function eg13(){

let x = [10,20,30];

let y = [40,5,60,70];

let z = [...x, 100, 200, ...y];

for(let a of z) console.log(a);

}

eg13();

function eg14\_1(...data:number[]){

console.log("eg14\_1 with ...data:number[] paraamter got executed");

for (let a of data) console.log(a);

}

function eg14\_2(data:number[]){

console.log("eg14\_2 with data:number[] parameter goot executed");

for (let x of data) console.log(x);

}

function eg14(){

let x = [10,20,30,40];

eg14\_1(...x);

eg14\_2(x);

//eg14\_1(x); // incorrect

//eg14\_2(10,20,30); // incorrect

}

eg14();