Advanced Problem Solving: Conditional Statements, Iterative Statements, Functions

Relevel
by Unacademy

Topics Covered

Advanced programming problem in JavaScript around

Conditional Statements

Iterative Statements

Functions

Multiple Choice Questions

Practice Questions

Conditional Statements

1. If/else statement

- The **if** statement is executed if a specified condition is truthy, otherwise the condition is treated as falsy and another statement is executed.
- In JavaScript, a **truthy** and **falsy** are the values that are considered true or false respectively when encountered in a boolean context.

Syntax

```
if (condition)
     { statement1 }
else if
     { statement2 }
...
else
     { statementn }
```

Conditional Statements

Problem 1 –

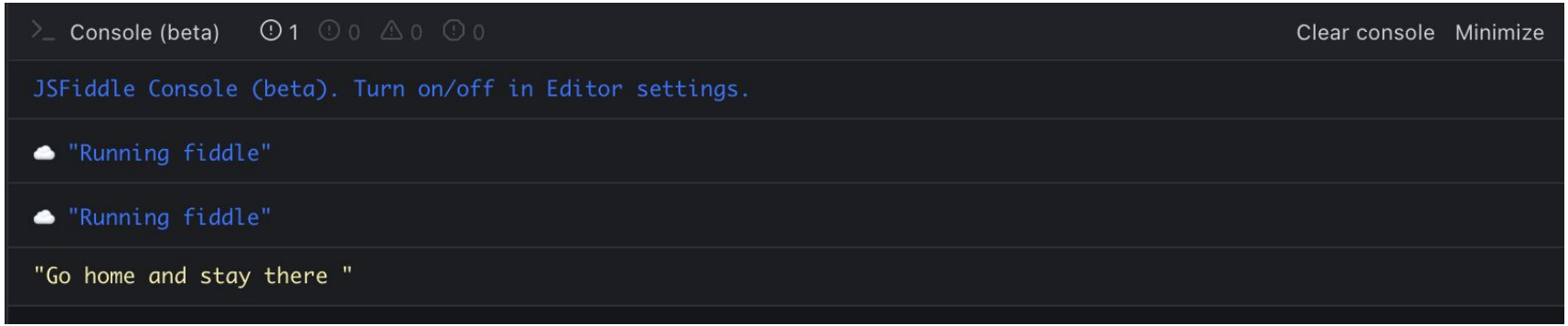
Create a program which gives output for children to go out in park if the temperature is between 20 degrees Celsius – 25 degrees Celsius and if it's not raining outside, ask them to be in play school if the temperature is between 18 degrees Celsius to 20 degrees Celsius and raining otherwise, they should not step out of the home.

Code Snippet of solution:

JavaScript + No-Library (pure JS) ▼ Tidy

```javascript
let temp = 20;
let isRaining = true;

if ((temp >= 20 && temp <= 25) && isRaining == false) {
  console.log("Yay! you can go out to play ");
} else if ((temp >= 18 && temp <= 20) && isRaining == false) {
  console.log("Stay inside the play school");
} else {
  console.log("Go home and stay there ");
}

```

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"Running fiddle"
"Go home and stay there "

Conditional Statements

2. Ternary Operator

- Frequently used as a substitute for if...else statement, this is the only operator that takes 3 operands.

Syntax :

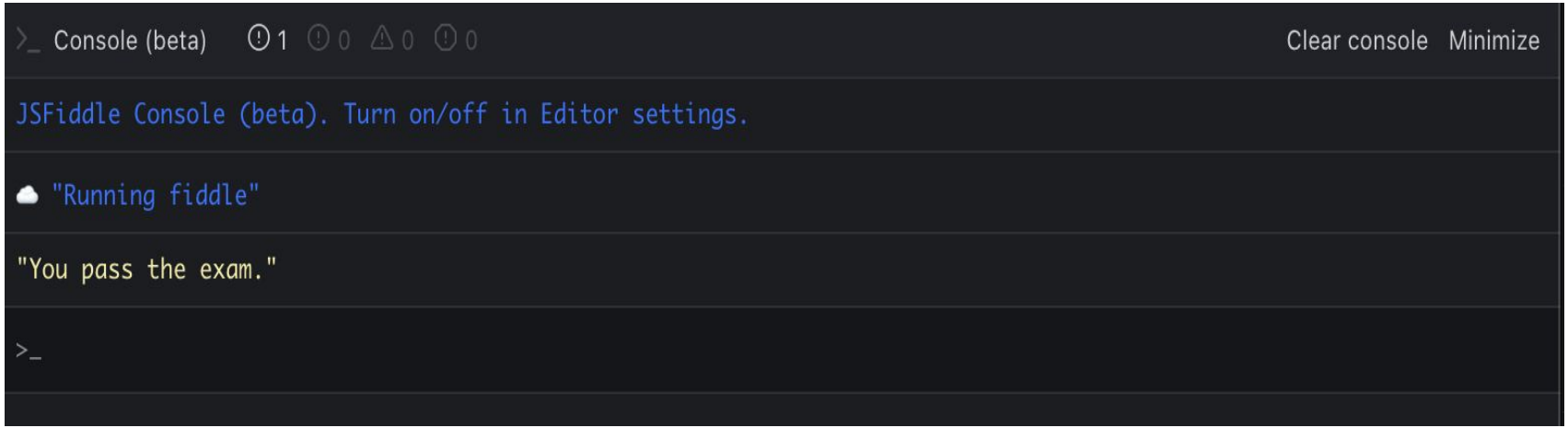
```
condition ? exprIfTrue : exprIfFalse
```

Conditional Statements

Problem 2 –

Create a program to print the result as pass if the marks obtained is greater than or equal to 40 using ternary operator.

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"You pass the exam."
>_

Conditional Statements

Problem 2 –

Create a program to print the result as pass if the marks obtained is greater than or equal to 40 using ternary operator.

Code Snippet of solution :

JavaScript + No-Library (pure JS) ▼

```
// program to check pass or fail

let obtained_markes = 46

// check the condition
let result_value = (obtained_markes >= 40) ? 'pass' : 'fail';

console.log(`You ${result_value} the exam.`);
```

Conditional Statements

3. Switch Case

Frequently used to perform various actions based on the various conditions being passed

Syntax

```
switch(expressions) {
      case 1:
      // code
      break
     case 2:
      //code
      break
     default;
}
```

Conditional Statements

Problem 3 –

Create a program to find if the number is positive, negative or zero.

Solution –

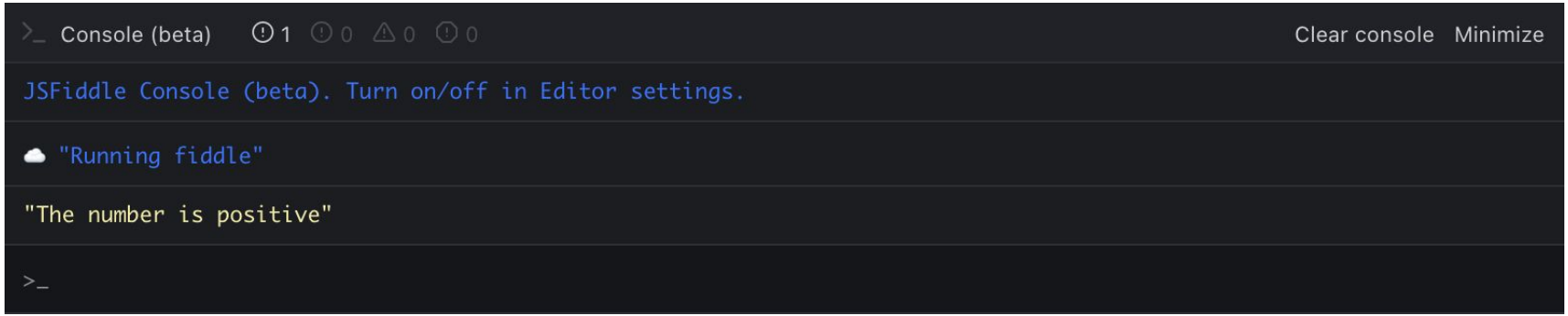
Check for condition with 0 with a switch case.

Code Snippet of solution :

JavaScript + No-Library (pure JS) ▼

```javascript
// program that checks if the number is positive, negative or zero
const checked_number = 5;

let x = 4;

switch (true) {
  case (x > 0):
    text = "The number is positive";
    break;
  case (x === 0):
    text = "The number is zero";
    break;
    case (x < 0):
    text = "The number is negative";
    break;
  default:
    text = "No value found";
}
console.log(text);

```

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"The number is positive"
>_

Iterative Statements

1. for Statement

- When a specific code block is to be re-executed for a certain number of times based on specific conditions, we use iterative statements.

Syntax

```
for(condition1; condition2; condition3) {
    // code
}
for(key in object) {
    // code
}
```

Iterative Statements

Problem 4 –

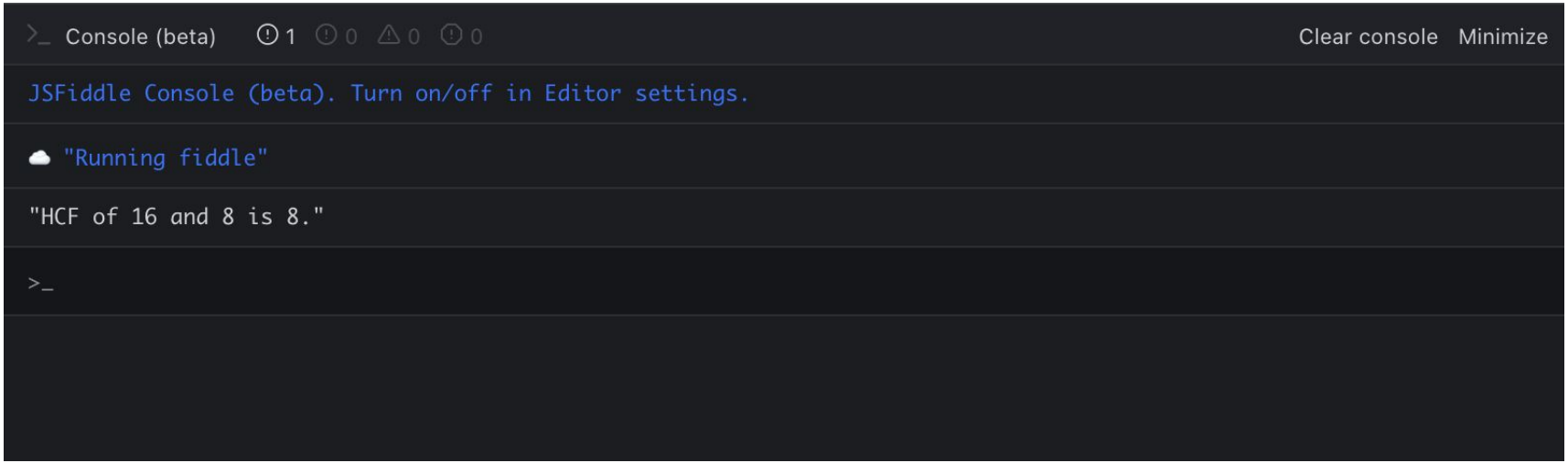
Write a program to find the HCF or GCD of two integers

Code Snippet of solution :

JavaScript + No-Library (pure JS) ▼

```javascript
// program to find the HCF or GCD of two integers

let hcf_value;
// take input
const first_number = 16;
const second_number = 8;

// looping from 1 to number1 and number2
for (let iterator = 1; iterator <= first_number && iterator <= second_number; iterator++) {

    // check if is factor of both integers
    if( first_number % iterator == 0 && second_number % iterator == 0) {
        hcf_value = iterator;
    }
}

// display the hcf
console.log(`HCF of ${first_number} and ${second_number} is ${hcf_value}.`);
```

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"HCF of 16 and 8 is 8."
>_

Functions

A function is a set of statements that takes an inputs, perform the relevant computation based on the statement written. Furthermore, functions are first-class objects, because they can have properties and methods just like any other object

Syntax

We can invoke a function in JavaScript in the following ways:

- Function as a statement.
- Function as an expression.
- Function as an arrow function.
- Function using the Function constructor.

Problem 5 –

Create a program to reverse a string.

Solution –

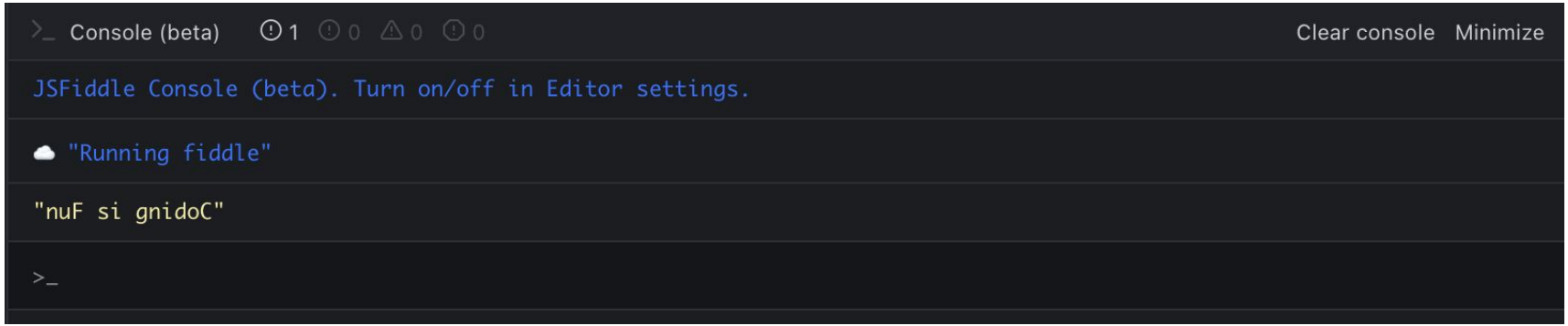
Traverse the input string from last index and add each character to a new string. Print the new reversed string.

Code Snippet of solution :

JavaScript + No-Library (pure JS) ▼

```javascript
// program to reverse a string

function reversingString(string_value) {

    // empty string
    let updated_string = "";
    for (let iterator = string_value.length - 1; iterator >= 0; iterator--) {
        updated_string += string_value[iterator];
    }
    return updated_string;
}

// take input from the user
const input_string = "Coding is Fun";

const output = reversingString(input_string);
console.log(output);
```

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"nuF si gnidoC"
>_

Problem 6 –

Create a program to find the largest amongst given 3 numbers.

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼ Tidy

```
// program to find the largest among three numbers

const first_number = 56;
const second_number = 84;
const thrid_number = 15;
let largest_number;

// check the condition
if(first_number >= second_number && first_number >= thrid_number) {
    largest_number = first_number;
}
else if (second_number >= first_number && second_number >= thrid_number) {
    largest_number = second_number;
}
else {
    largest_number = thrid_number;
}

// display the result
console.log("The largest value of number is " + largest_number);
```

Functions

Problem 7 –

Create a program to get sum of all digits of a number.

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼ Tidy

```javascript
/* Function to get sum of digits */
function generateSumOfDigits(number_passed)
{
    var output = 0;
    while (number_passed != 0) {
        output = output + number_passed % 10;
        number_passed = parseInt(number_passed / 10);
    }
    return output;
}

// Driver code
var value = 687;
console.log(generateSumOfDigits(value))
```

Problem 8 –

Find if a given number is an Armstrong number or not.

Hint : Armstrong number is a number that is equal to the sum of cubes of its digits

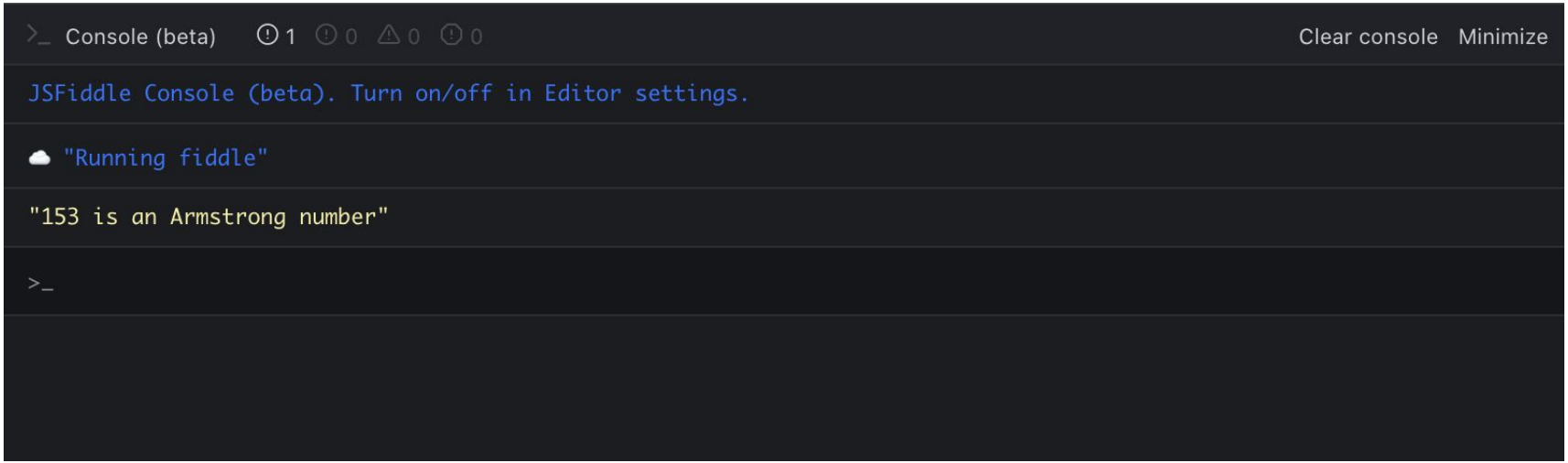
Solution Approach -

Find all the digits of the number (in our example find the ones, tens and hundreds place digit) and then find their cubes and add them to validate against the original number.

JavaScript + No-Library (pure JS) ▼ Tidy

```javascript
// program to check an Armstrong number of three digits

let output_value = 0;
const input_value = 153;

// create a temporary variable
let temporary_value = input_value;
while (temporary_value > 0) {
    // finding the one's digit
    let remainder_value = temporary_value % 10;

    output_value += remainder_value * remainder_value * remainder_value;

    // removing last digit from the number
    temporary_value = parseInt(temporary_value / 10); // convert float into integer
}
// check the condition
if (output_value == input_value) {
    console.log(`${input_value} is an Armstrong number`);
}
else {
    console.log(`${input_value} is not an Armstrong number.`);
}
```

Output :


Console (beta)
1
0
0
0
Clear console
Minimize
JSFiddle Console (beta). Turn on/off in Editor settings.
"Running fiddle"
"153 is an Armstrong number"
>_

Problem 9 –

Create a pattern using '*' as shown below using loops.

*

**

***

****

*****

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼ Tidy

```
let number_of_rows = 5; // the length of the grid
let star_string = "";
for (let row = 0; row < number_of_rows; row++) {
  for (let column = 0; column <= row; column++) {
    star_string += "*";
  }
  star_string += "<br>";
}
document.write("", star_string);
```

Problem 10 –

Create a pattern using '*' as shown below using loops.

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼ Tidy

```
let number_of_row = 5; // you can take input from prompt or change the value
let output_string = "";
// External loop
for (let row_first = 1; row_first <= number_of_row; row_first++) {
  // printing spaces
  for (let columns = number_of_row; columns > row_first; columns--) {
    output_string += " ";
  }
  // printing star
  for (let space_bet = 0; space_bet < row_first * 2 - 1; space_bet++) {
    output_string += "*";
  }
  output_string += "<br>";
}
document.write(`<pre>${output_string}</pre>`);
```

Problem 11 –

Create Fibonacci series of 5 and 8.

Hint : The Fibonacci sequence is a series of numbers in which each number is the sum of the two that precede it. Starting at 0 and 1, the sequence looks like this: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so on forever.

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼

```
1  //loop pattern problem 2/*
2  let total_num_of_rows = 5;
3  let printed_string = "";
4  // External loop
5  for (let row = 1; row <= total_num_of_rows; row++) {
6    // printing spaces
7    for (let column = 1; column <= total_num_of_rows - row; column++) {
8      printed_string += " ";
9    }
10   // printing star
11   for (let space = 0; space < 2 * row - 1; space++) {
12     printed_string += "*";
13   }
14   printed_string += "\n";
15 }
16 console.log(printed_string);
17
```

Problem 12 –

Write power function for any positive integer

Hint : Power function F(x^n) = x*x*x...n times; where x is the base and n is the exponent

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼ Tidy

```
function power(base, exponent) {
  var result = 1;
  if(exponent == undefined)
    exponent = 2;
  for(var i=1; i<=exponent; i++) {
    result = result * base;
  }
  return result;
}

document.write(power(2,4));
```

Problem 13 –

Print all the prime number up to a given number value.

Code Snippet of solution

JavaScript + No-Library (pure JS) ▼

```
var number_count = 10;
for (var prime_iterator = 2; prime_iterator <= number_count; prime_iterator++) {
  var isPrime = true;
  for (var divident = 2; divident <= Math.sqrt(prime_iterator); divident++) { // nested loop
    if (prime_iterator % divident === 0) {
      isPrime = false;
      break;
    }
  }
  if (isPrime) {
    document.write(prime_iterator + " ");
  }
}
document.write("<br>");
```

THANK YOU