

Assignment 3

Question 1

```
void main()

{

    List<String> ballList = ["Ball ()","Ball ()","Ball ()","Ball ()","Ball ()"];

    print(ballList);

    print("remove repeted element ");

    ballList.removeRange(0,4);

    print(ballList);

}
```

A screenshot of an IDE window. The left pane shows Java code with line numbers 1 through 22. The code defines a main method, initializes a List<String> named ballList with five "Ball ()" strings, prints the list, prints a message "remove repeted element ", removes the first five elements using removeRange(0,4), and prints the list again. The right pane is split into two sections: "Console" and "Documentation". The Console section shows the output of the program: "[Ball (), Ball (), Ball (), Ball (), Ball ()]", "remove repeted element", and "[Ball ()]". The Documentation section shows the JavaDoc for the List<E> class, stating it is an abstract class that implements EfficientLength and is an indexable collection of objects with a length.

Question 2

```
void main()

{

    List<int> one = [1,2,3,4,5,6,7];

}
```

```

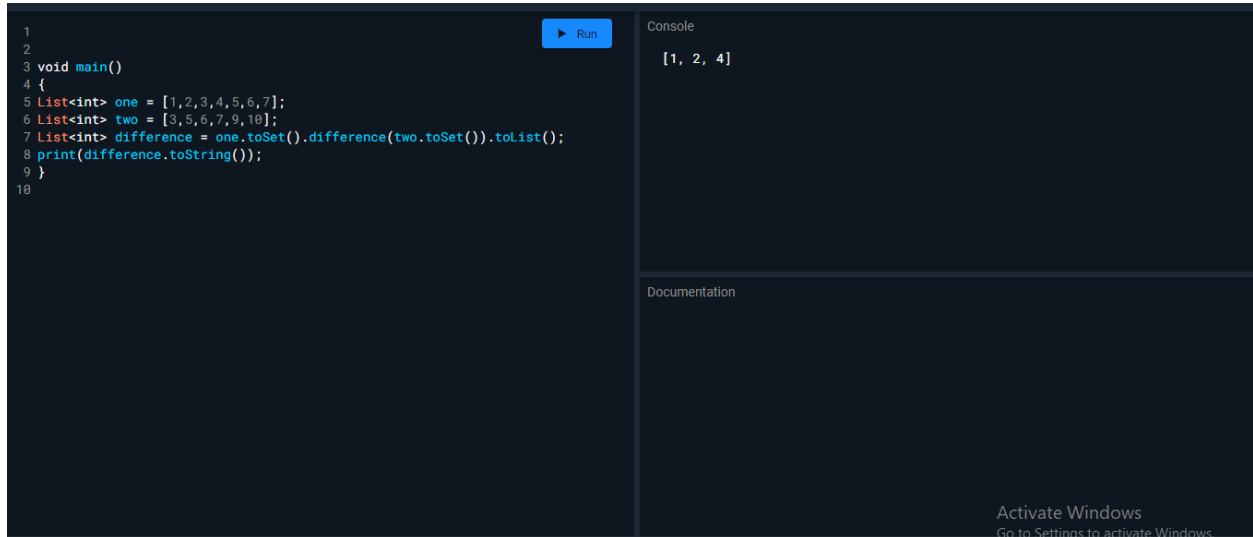
List<int> two = [3,5,6,7,9,10];

List<int> difference = one.toSet().difference(two.toSet()).toList();

print(difference.toString());

}

```



The screenshot shows a code editor with a dark theme. On the left, a Kotlin program is written in lines 1 through 10. Line 1 is empty. Line 2 is empty. Line 3 starts with 'void main()'. Line 4 starts with '{'. Line 5 defines 'one' as a list [1,2,3,4,5,6,7]. Line 6 defines 'two' as a list [3,5,6,7,9,10]. Line 7 calculates 'difference' as the set difference of 'one' and 'two'. Line 8 prints the difference. Line 9 closes the main function block with '}'. Line 10 is empty. A blue 'Run' button is located to the right of line 2. On the right side of the editor, there is a 'Console' panel showing the output '[1, 2, 4]'. Below the console is a 'Documentation' panel, which is currently empty. At the bottom right of the editor, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'

```

1
2
3 void main()
4 {
5   List<int> one = [1,2,3,4,5,6,7];
6   List<int> two = [3,5,6,7,9,10];
7   List<int> difference = one.toSet().difference(two.toSet()).toList();
8   print(difference.toString());
9 }
10

```

Run

Console

[1, 2, 4]

Documentation

Activate Windows
Go to Settings to activate Windows.

Question 3

```

void main() {

    List<int> a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100];

    int i = 0;

    List<int> l = [];

    for (var e in a) {

        if (++i % 2 == 0) {

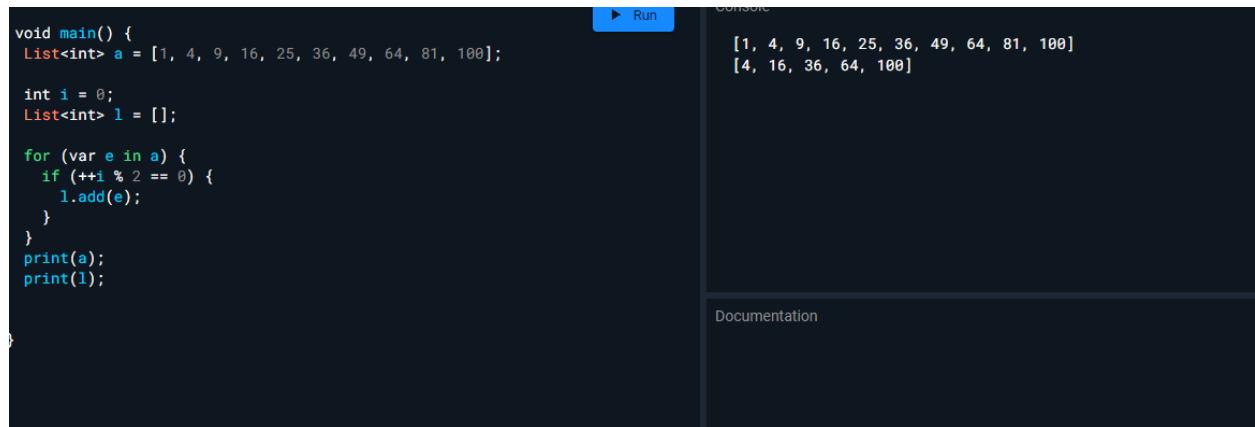
            l.add(e);

        }

    }
}

```

```
}  
  
print(a);  
  
print(l);  
  
}
```



The screenshot shows a C# IDE with a code editor on the left and a console on the right. The code in the editor defines a `main` method that creates an array `a` with values [1, 4, 9, 16, 25, 36, 49, 64, 81, 100], initializes an integer `i` to 0, and creates an empty `List<int>` `l`. A `for` loop iterates over the array `a`, and for every second element (where `i % 2 == 0`), it adds the element to the list `l`. After the loop, it prints the array `a` and the list `l`. The console on the right shows the output of these print statements: the first line is the array `[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]` and the second line is the list `[4, 16, 36, 64, 100]`. A "Run" button is visible above the console.

```
void main() {  
    List<int> a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100];  
  
    int i = 0;  
    List<int> l = [];  
  
    for (var e in a) {  
        if (++i % 2 == 0) {  
            l.add(e);  
        }  
    }  
    print(a);  
    print(l);  
}
```

Console

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]  
[4, 16, 36, 64, 100]
```

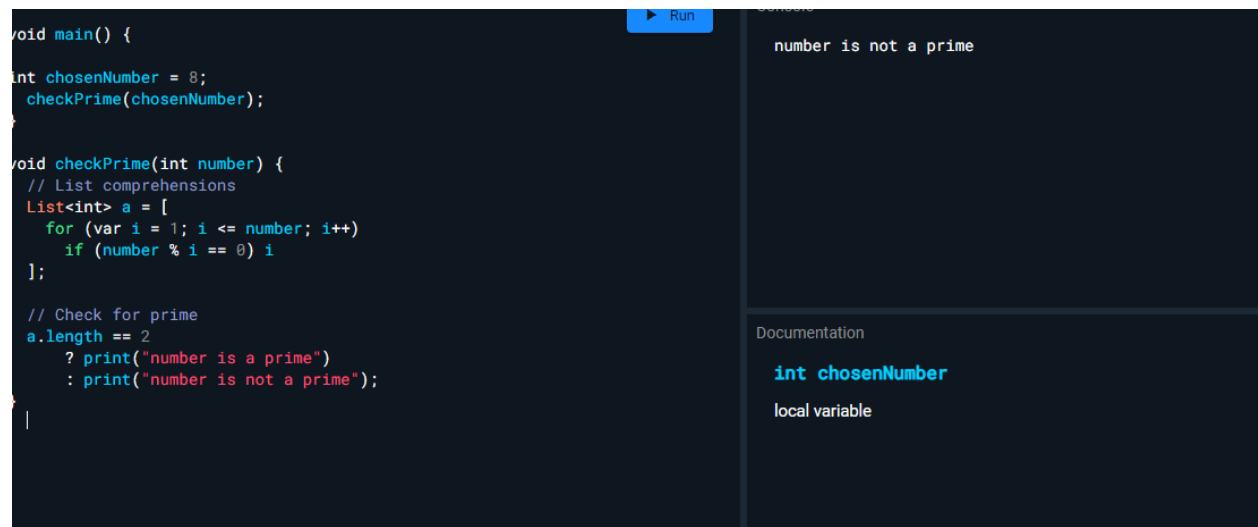
Documentation

Question 4

```
void main() {  
  
  
    int chosenNumber = 8;  
  
    checkPrime(chosenNumber);  
}
```

```
}
```

```
void checkPrime(int number) {  
  
    // List comprehensions  
  
    List<int> a = [  
  
        for (var i = 1; i <= number; i++)  
  
            if (number % i == 0) i  
  
    ];  
  
  
    // Check for prime  
  
    a.length == 2  
  
        ? print("number is a prime")  
  
        : print("number is not a prime");  
  
}
```



```
void main() {  
    int chosenNumber = 8;  
    checkPrime(chosenNumber);  
}  
  
void checkPrime(int number) {  
    // List comprehensions  
    List<int> a = [  
        for (var i = 1; i <= number; i++)  
            if (number % i == 0) i  
    ];  
  
    // Check for prime  
    a.length == 2  
        ? print("number is a prime")  
        : print("number is not a prime");  
}
```

number is not a prime

Documentation

int chosenNumber

local variable

```
void main() {
    int chosenNumber = 7;
    checkPrime(chosenNumber);
}

void checkPrime(int number) {
    // List comprehensions
    List<int> a = [
        for (var i = 1; i <= number; i++)
            if (number % i == 0) i
    ];

    // Check for prime
    a.length == 2
    ? print("number is a prime")
    : print("number is not a prime");
}
```

Run

Console

number is a prime

Documentation

Question 5

```
void main(){

    print("multiplication table of 7 length 15");

    for(var i=0 ; i< 105; )

    {

        i= i+6;

        i++;

        print(" ${i}");

    }

}
```

```
1
2
3
4 void main(){
5
6   print("multiplication table of 7 length 15");
7   for(var i=0 ; i< 105; )
8
9   {
10
11     i= i+6;
12     i++;
13     print(" ${i}");
14
15   }
16
17 }
18
19
20
21
22
23
24
```

Run

multiplication table of 7 length 15
7
14
21
28
35
42
49
56
63
70
77

Documentation

Question 6

```
void main(){
```

```
var fruits = ['apple', 'banana', 'mango', 'orange', 'strawberry'];
```

```
for(var i=0;i<fruits.length;i++)
```

```
{
```

```
    print(fruits[i]);
```

```
}  
  
}
```

```
void main(){  
    var fruits = ['apple', 'banana', 'mango', 'orange', 'strawberry'];  
    for(var i=0;i<fruits.length;i++){  
        print(fruits[i]);  
    }  
}
```

```
apple  
banana  
mango  
orange  
strawberry
```

Documentation

abstract class int extends num

An integer number.

The default implementation of **int** is 64-bit two's complement integers with operation wrap to that range on overflow.

Note: When compiling to JavaScript, integers are restricted to values that can be represented by JavaScript's Number type.

Question 7

```
void main()  
  
{  
  
    var i =0;  
  
    while (i * 5 < 100)  
    {  
        i = i + 1;  
        print (i * 5);  
    }  
  
}
```

Question 8

```
void main(){

    //CONVERT CALCIUS IS FAHRENHEIT

    var cal = 35;

    var F = (cal * 9) / 5 + 32;

    print("NNoC is NNoF: ${F}");

    //CONVERT FAHRENHEIT IS CALCIUS

    var farh = 95;

    var C = (farh - 32) * 5 / 9;

    print("NNoF is NNoC: ${C}");

}
```



```
void main(){  
    //CONVERT CALCIUS IS FAHRENHEIT  
    var cal = 35;  
  
    var F = (cal * 9) / 5 + 32;  
  
    print("NNoC is NNoF: #{F}");  
  
    //CONVERT FAHRENHEIT IS CALCIUS  
    var farh = 95;  
  
    var C = (farh - 32) * 5 / 9;  
  
    print("NNoF is NNoC: #{C}");  
|  
}
```

Run

Console

```
NNoC is NNoF: 95  
NNoF is NNoC: 35
```

Documentation

Question 9

```
void main(){
```

```
var user1= 10;
```

```
var user2 = 5;
```

```
//print("press 1 to add a number");
```

```
//print("press 2 to subtract a number");
```

```
//print("press 3 to multiply a number");
```

```
//print("press 4 to divide a number");
```

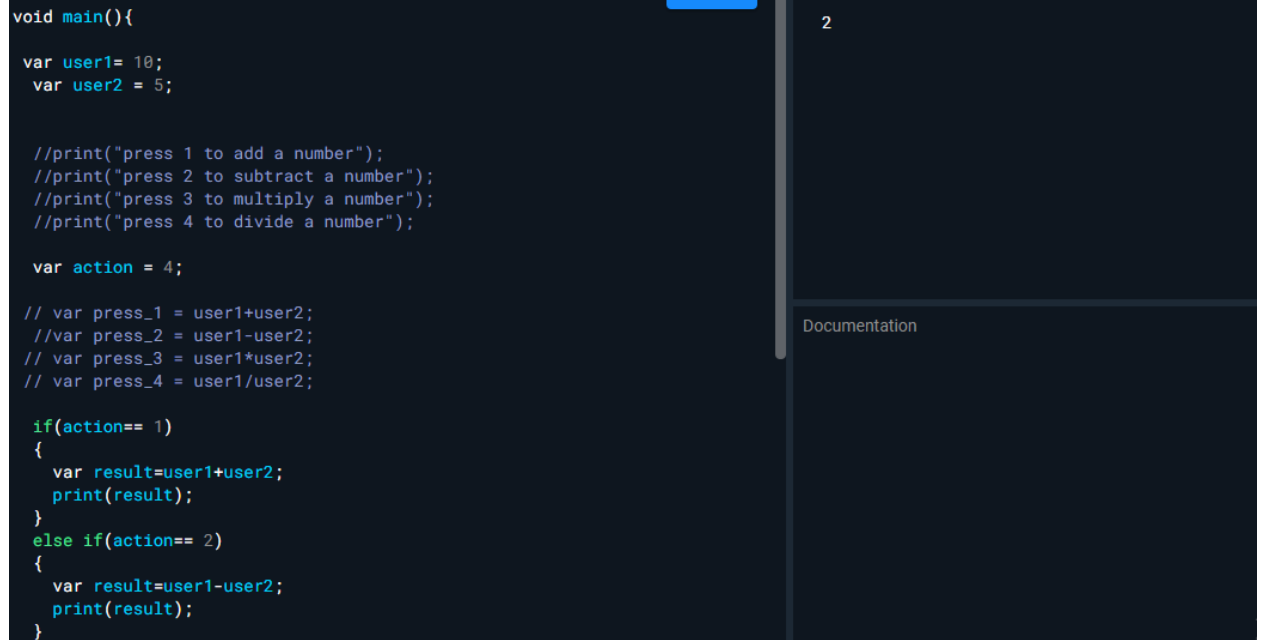
```
var action = 4;
```

```
// var press_1 = user1+user2;
```

```
//var press_2 = user1-user2;  
// var press_3 = user1*user2;  
// var press_4 = user1/user2;
```

```
if(action== 1)  
{  
    var result=user1+user2;  
    print(result);  
}  
else if(action== 2)  
{  
    var result=user1-user2;  
    print(result);  
}  
else if(action== 3)  
{  
    var result=user1*user2;  
    print(result);  
}  
else if(action== 4)  
{  
    var result=user1/user2;  
    print(result);  
}
```

```
else
{
    print("wrong input");
}
}
```



```
void main(){
    var user1= 10;
    var user2 = 5;

    //print("press 1 to add a number");
    //print("press 2 to subtract a number");
    //print("press 3 to multiply a number");
    //print("press 4 to divide a number");

    var action = 4;

    // var press_1 = user1+user2;
    //var press_2 = user1-user2;
    // var press_3 = user1*user2;
    // var press_4 = user1/user2;

    if(action== 1)
    {
        var result=user1+user2;
        print(result);
    }
    else if(action== 2)
    {
        var result=user1-user2;
        print(result);
    }
}
```

2

Documentation

```

void main(){

var user1= 10;
var user2 = 5;

//print("press 1 to add a number");
//print("press 2 to subtract a number");
//print("press 3 to multiply a number");
//print("press 4 to divide a number");

var action = 7;

// var press_1 = user1+user2;
//var press_2 = user1-user2;
// var press_3 = user1*user2;
// var press_4 = user1/user2;

if(action== 1)
{
var result=user1+user2;
print(result);
}
else if(action== 2)
{
var result=user1-user2;
print(result);
}
}

```

Run

Console

wrong input

Documentation

```
void main()
```

```
{
```

```
//check and compare alphabets in this list
```

```
var vowels = ['a', 'e', 'i', 'o', 'u'];
```

```
var alphabet = 'i';
```

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

```
if(alphabet == vowels[i])
```

```

        // if (vowels[i] == 'a' || vowels[i] == 'e' || vowels[i] == 'i' || vowels[i] == 'o' || vowels[i] ==
'u' || vowels[i] == 'e' ) {

            {

                print("vowels");

            }

        else{

            print("is not a vowel");

        }

    }

}

```

The screenshot shows a code editor with a Go program and its console output. The code defines a list of vowels and iterates through them, printing each one. The console output shows the vowels being printed, followed by an error message: "Uncaught Error: RangeError (index): Index out of range: index should be less than 5: 5".

```

1 void main()
2 {
3
4 //check and compare alphabets in this list
5 var vowels = ['a', 'e', 'i', 'o', 'u'];
6
7 var alphabet = 'i';
8
9 for (var i = 0; i <= vowels.length; i++){
10
11     if(alphabet == vowels[i])
12         // if (vowels[i] == 'a' || vowels[i] == 'e' || vowels[i] == 'i' || vowels[i] == 'o' || vowels[i] == 'u') {
13         {
14             print("vowels");
15         }
16     else{
17         print("is not a vowel");
18     }
19 }
20 }
21
22 |
23
24
25

```

Console Output:

```

is not a vowel
is not a vowel
vowels
is not a vowel
is not a vowel
is not a vowel
Uncaught Error: RangeError (index): Index out of range: index should be less than 5: 5

```

Documentation