LMS, Recording, Assignment

Name, exp, language, Java JS Java JS Java Java C++ Python Java Java

Github Repo

https://github.com/leangaurav/DSA_2023_01_29 (https://github.com/leangaurav/DSA_2023_01_29)

Some math

1. Sum of first n numbers
2. Mean, median, mode
3. Permutation Combinations
4. Modulo arithmetic
   (A + B) mod C = (A mod C + B mod C) mod C
   (A * B) mod C = (A mod C * B mod C) mod C
   A^B mod C = ((A mod C)^B) mod C

- Sum first n natural numbers: (n * (n+1))/2
- Mean: Average (Sum of all no.s) / (number of elements)
- Median: Arrange data in ascending order: 50th percentile
  - N is even: sum of middle two elements / 2
  - N is odd: there's a middle element
- Mode: most frequent element in data

In [ ]:

```
```

```
## Common Terms
1. Pallindrome
   Reverse and original is same:
   12321 - y
   123 - n
   1221 - y
   abcba - y
   abba -y
   ab - N
   a - y

1. Factorial
   5! =  5 * 4 * 3 * 2 * 1
   N! = 1 * 2 * ... N-1 * N

1. Fibonacci Series
   current element is sum of previous two elements

   0 , 1 , 1 , 2, 3, 5 ,8 ,13......

1. Sorting
   data arragned in asc or desc order

   10, 4, 5 , 11
   4,5,10,10,11
```

In [ ]:

```
a->97, b->98...
A->65
```

In [4]:

```
print('a' > 'b')
print('a' > 'A')
```

```
False
True
```

In [6]:

```
a = ["apple", "ace", "pqr", "ball", "XYZ"]
a.sort()
print(a)
```

```
['XYZ', 'ace', 'apple', 'ball', 'pqr']
```

In [ ]:

In [ ]:

If else

1. Different constructs
2. Ternary op
3. find max of 2 numbers
4. find max of 3 numbers
5. Built-in min-max

WAP to find the maximum of 2 numbers

```
// V1
int n1, n2;
/// input values of n1, n2

if (n1 > n2)
    cout << n1;
else {
    cout << n2;
}



// V2
int n1, n2;
/// input values of n1, n2

int max = n1;
if (n2 > max) {
    max = n2;
}

cout << max;


if (n1 > n2)
    cout << n1;
else {
    cout << n2;
}
```

if-else if-else

WAP to find max 3 numbers

```
int n1, n2, n3;

// input values of n1, n2, n3

if (n1 >= n2 && n1 >= n3) {
    cout << n1;
} else if (n2 >= n1 && n2 >= n3) {
    cout << n2;
} else {
    cout << n3;
}


// V2
int max = n1;
if (n2 > max) {
    max = n2;
}
if (n3 > max) {
    max = n3;
}
cout << max;


if (n1 >= n2) {
    if (n1 >= n3)
        cout << n1;
    else
        cout << n3;
} else {
    if (n2 >= n3)
        cout << n2;
    else
        cout << n3;
}
```

In [ ]:

Loops

1. Patterns
2. Factorial
3. Fibonacci series

for, while, do-while

```
for( init; condition ; update ) {

}


#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i < 10; i++) {
        cout << i;
    }
    return 0;
}

// i 0 1 2 3 4 5 6 7 8 9 10


0123456789
```

```
for (int i =0 ; i < 5; i++) {
    cout << "*";
}
*****
```

```
for (int i = 1; i <= 5; i++) {
    cout << "*";
}
*****
```

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        cout << "*";
    }
}
*************************
```

```
// i 1 1 1 1 1 - 2
// j 1 2 3 4 5 6 1 2 3 4 5...

for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        cout << "*";
        cout << "\n";
    }
}
```

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        cout << "*";
    }
    cout << "\n";
}
*****
*****
*****
*****
*****
```

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        cout << j;
    }
    cout << "\n";
}
12345
12345
12345
12345
12345
```

```
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= i; j++) {
        cout << j;
    }
    cout << "\n";
}
1
12
123
1234
12345
```

```
for (int i = 5; i >= 1; i--) {
    for (int j = 1; j <= i; j++) {
        cout << j;
    }
    cout << "\n";
}
12345
1234
123
12
1


for (int i = 1; i <= 5; i++) {
    for (int j = 5-i; j >= 1; j--) {
        cout << " ";
    }
    for (int j = 1; j <= i; j++) {
        cout << j;
    }
    cout << "\n";
}
    1
   12
  123
 1234
12345

for (int i = 1; i <= 5; i++) {
    for (int j = 5-i; j >= 1; j--) {
        cout << " ";
    }
    for (int j = 1; j <= i*2-1; j++) {
        cout << j;
    }
    cout << "\n";
}

    1
   123
  12345
 1234567
123456789



for (int i = 1; i <= 5; i++) {
    for (int j = 5-i; j >= 1; j--) {
        cout << " ";
    }

    for (int j = 1; j <= i; j++) {
        cout << j;
    }

    for (int j = i-1; j >= 1; j--) {
        cout << j;
    }
    cout << "\n";
}

    1
   121
  12321
 1234321
123454321
```

```
In [ ]:
```

Functions

In [ ]:

A piece of code: name

```
void printArray(int arr[], int n) {
    for (int i = 0; i < n; i+=1) {
        cout << arr[i];
    }
}

int main() {
    int arr[] = {1,2,3,4,5};
    printArray(arr);
    printArray(arr);
}
```

- modular
- reuse

In [ ]:

- Are we going to have complete course in C++ or Python ?
  Neither - C++, Python
- Will you be teaching time complexity and space complexity today/tomorrow?
  Next week Saturday

In [ ]:

https://leetcode.com/problems/fizz-buzz/description/ (https://leetcode.com/problems/fizz-buzz/description/)

In [ ]:

```
class Solution:
    def fizzBuzz(self, n: int) -> List[str]:
        ans = []

        for i in range(1,n+1):
            if(i%3 == 0  and i%5 == 0):
                ans.append("FizzBuzz")
            elif(i%3 == 0):
                ans.append("Fizz")
            elif(i%5 == 0):
                ans.append("Buzz")
            else:
                ans.append(str(i))
        return ans
```

In [ ]:

```
class Solution {
    public List<String> fizzBuzz(int n) {
        List<String> result = new ArrayList<>();

        for (int i=1; i<=n; i++){
            String value = String.valueOf(i);
            if(i%3 == 0 && i%5 == 0){
                value = "FizzBuzz";
            }else if(i%3 == 0){
                value = "Fizz";
            }else if(i%5 == 0){
                value = "Buzz";
            }

            result.add(value);
        }

        return result;

    }
}
```