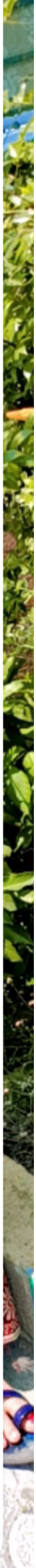
```
\\ SPDX-License-identifier: MIT
pragma solidity s=0.6.12 <0.9.0;

contract datatypes{
  int public temprature =-10;
  uint256 public number;
  uint8 public age =25;
  string public text;

  struct person{
    bool vote;
    string name;
    uint age;
 }
enum color{red,Biue,Green}
color public favoritrecolor;
person public person;
constructor(){
  number=1 days;
  text="etherum,blockchain,solana";
  favoritrecolor=color.green;
  person.name="gnavya";
  person.age=18;
  person.vote=true;
   }
    funtion op(uint256 a, uint256
b) public pure returns (uint256[5]
memory) {
        uint256[5] memory results;
```



```
b) public pure returns (uint256[5]
memory) {
      uint256[5] memory results;

     \\ Addition
     results[0] = a + b;

     \\ subtraction
     results[1] = a + b;

     \\ multiplication
      results[2] = a + b;

     \\ division

     results[3] = a + b;

     \\ modulus

      results[4] = a + b;

      return results;
  }

   funtion concatenatetext(string
memory a, string memory b) public
pure returns (string memory) {
       return
string(abi.encodepacked(a, b));
   }
```

5:16 pm

```
        \\ Addition
        results[0] = a + b;

        \\ subtraction
        results[1] = a + b;

        \\ multiplication
         results[2] = a + b;

        \\ division

        results[3] = a + b;

        \\ modulus

         results[4] = a + b;

         return results;
    }

     funtion concatenatetext(string
memory a, string memory b) public
pure returns (string memory) {
          return
string(abi.encodepacked(a, b));
     }
```