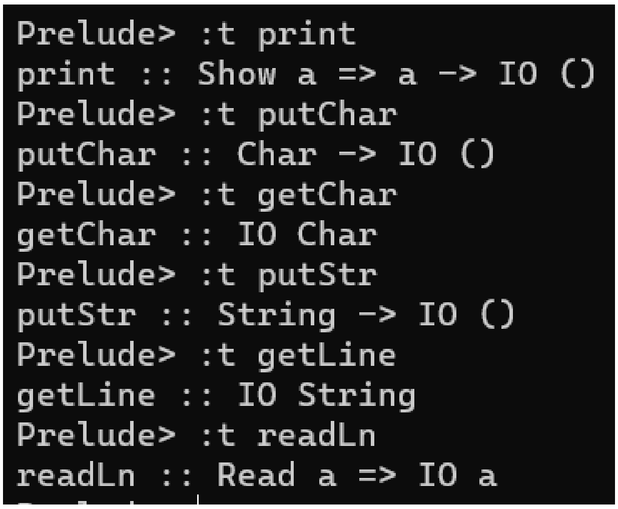
19CSE313

Principles of Programming Languages

Lab 9

S Abhishek

AM.EN.U4CSE19147

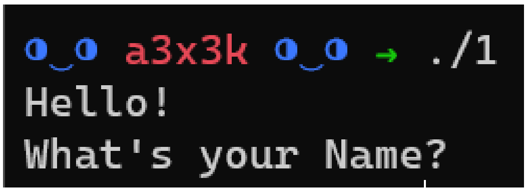


```
Prelude> :t print
print :: Show a => a -> IO ()
Prelude> :t putChar
putChar :: Char -> IO ()
Prelude> :t getChar
getChar :: IO Char
Prelude> :t putStr
putStr :: String -> IO ()
Prelude> :t getLine
getLine :: IO String
Prelude> :t readLn
readLn :: Read a => IO a
```

1 - Using do command

```
main = do
{
        putStrLn "Hello!";
        putStrLn "What's your Name?";
}
```

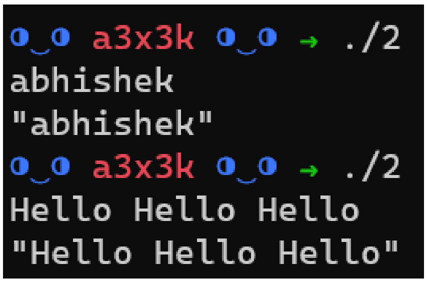
```
main = do
        putStrLn "Hello"
        putStrLn "What's your Name?"
```



```
◐‿◐ a3x3k ◐‿◐ ➜ ./1
Hello!
What's your Name?
```

2 - To create a list of characters reading the input from the user.

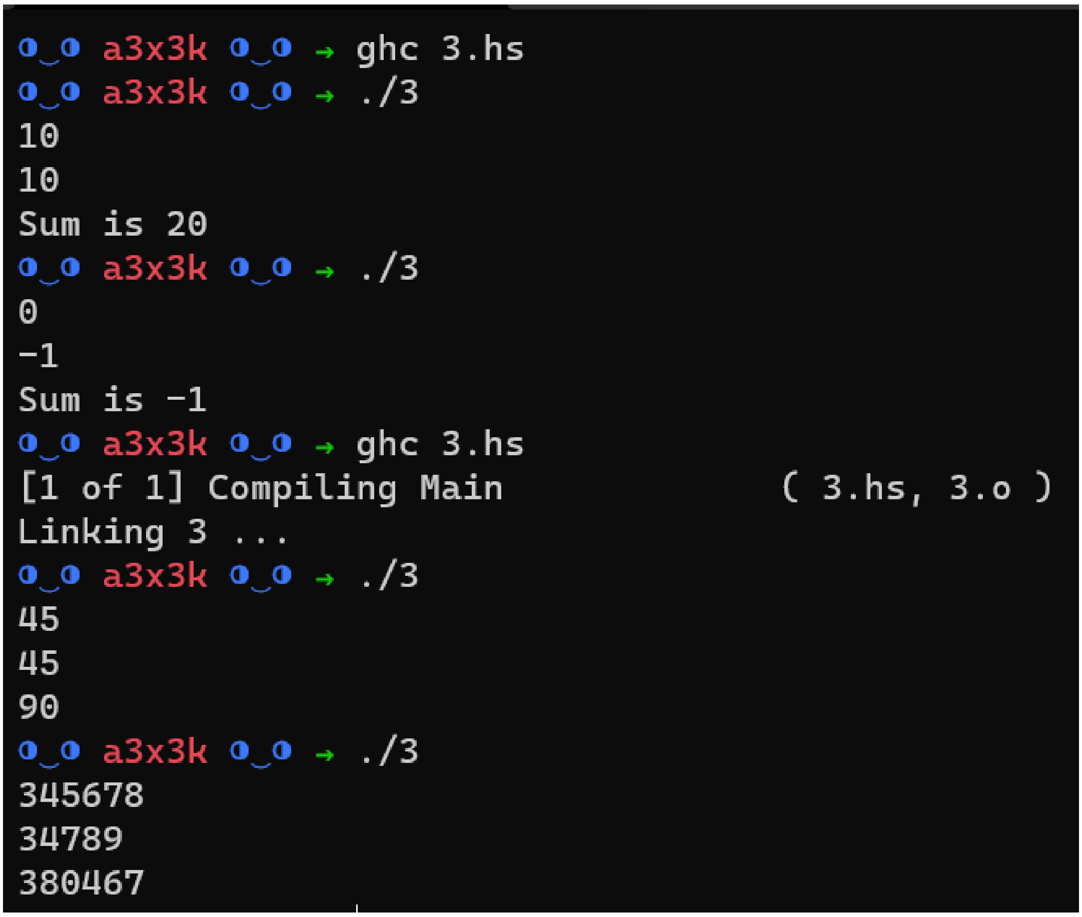
```
main = do

{

        a <- getL;

        print a;

}


getL :: IO [Char]

getL = do {

c <- getChar ;

if c == '\n' then return [];

else do {

 cs <- getL;

 return (c : cs); }}
```



```
◐‿◐ a3x3k ◐‿◐ ➜ ./2
abhishek
"abhishek"
◐‿◐ a3x3k ◐‿◐ ➜ ./2
Hello Hello Hello
"Hello Hello Hello"
```

3 - To find the sum of two numbers read from the user.

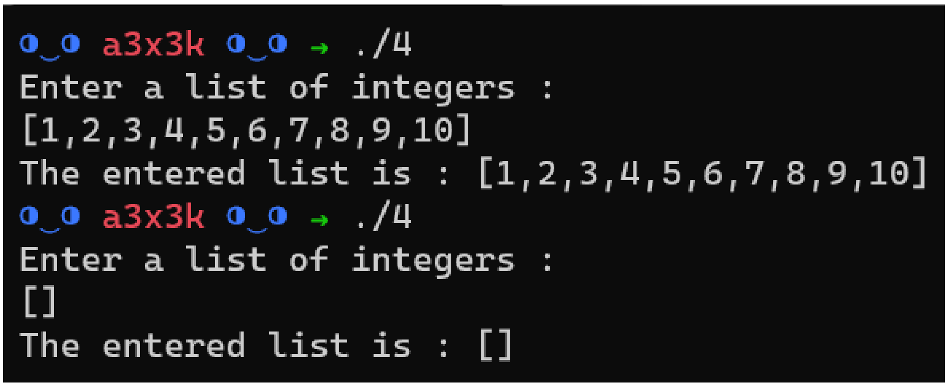
```
main = do
     a <- readLn
     b <- readLn
     putStr "Sum is "
     print (a+b)

mysum a b = a + b
main = do
     a <- readLn
     b <- readLn
     print $ mysum a b
```



```
o‿o a3x3k o‿o → ghc 3.hs
o‿o a3x3k o‿o → ./3
10
10
Sum is 20
o‿o a3x3k o‿o → ./3
0
-1
Sum is -1
o‿o a3x3k o‿o → ghc 3.hs
[1 of 1] Compiling Main             ( 3.hs, 3.o )
Linking 3 ...
o‿o a3x3k o‿o → ./3
45
45
90
o‿o a3x3k o‿o → ./3
345678
34789
380467
```

4 - To read a list of integers from the user and print it.

```
main = do

   putStrLn "Enter a list of integers : "

   list <- getLine

   putStr "The entered list is : "

   print (read list :: [Int])
```

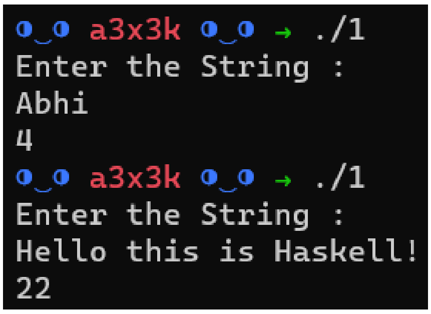


```
◑‿◑ a3x3k ◑‿◑ → ./4
Enter a list of integers :
[1,2,3,4,5,6,7,8,9,10]
The entered list is : [1,2,3,4,5,6,7,8,9,10]
◑‿◑ a3x3k ◑‿◑ → ./4
Enter a list of integers :
[]
The entered list is : []
```

Exercise

1 - Read a string and display its length.

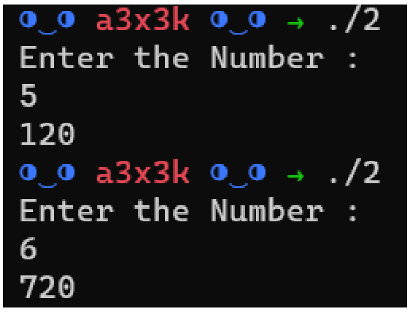
```
main = do

   putStrLn "Enter the String : "

   s <- getLine

   print (length(s))
```



```
◑‿◑ a3x3k ◑‿◑ → ./1
Enter the String :
Abhi
4
◑‿◑ a3x3k ◑‿◑ → ./1
Enter the String :
Hello this is Haskell!
22
```

2 - Read an integer and print the factorial of the number.

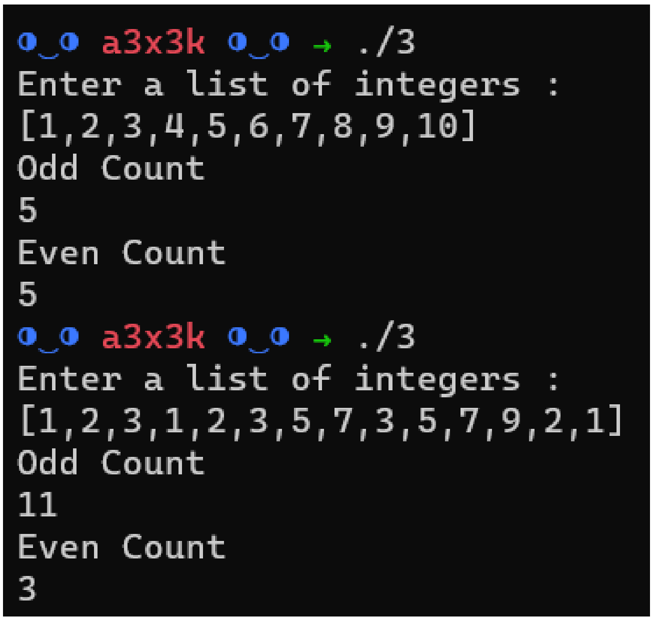
```
main = do
   putStrLn "Enter the Number :
   n <- readLn
   print (product [1..n])
```



```
◑‿◑ a3x3k ◑‿◑ → ./2
Enter the Number :
5
120
◑‿◑ a3x3k ◑‿◑ → ./2
Enter the Number :
6
720
```

3 - Display the count of even and odd numbers from the list of integers.

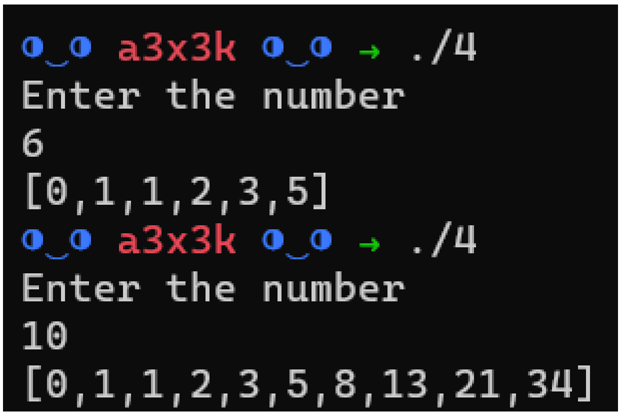
```
main = do
   putStrLn "Enter a list of integers : "
   list <- getLine
   let ls = read list :: [Int]
   putStrLn "Odd Count"
   print (length (filter odd ls))
   putStrLn "Even Count"
   print (length (filter even ls))
```



```
◑‿◑ a3x3k ◑‿◑ → ./3
Enter a list of integers :
[1,2,3,4,5,6,7,8,9,10]
Odd Count
5
Even Count
5
◑‿◑ a3x3k ◑‿◑ → ./3
Enter a list of integers :
[1,2,3,1,2,3,5,7,3,5,7,9,2,1]
Odd Count
11
Even Count
3
```

4 - Print n Fibonacci numbers

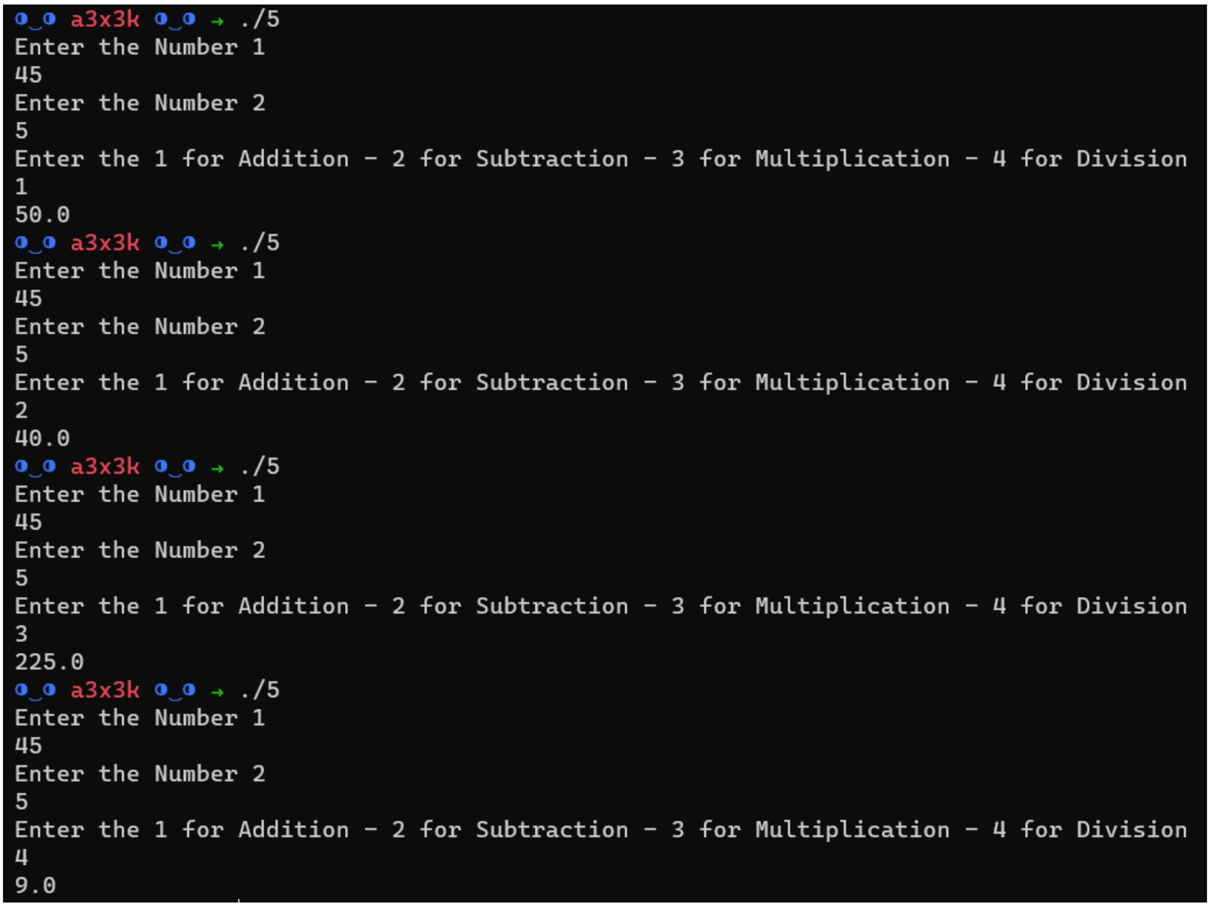
```
fib a b = a : fib b (a+b)

main = do

    putStrLn "Enter the number"

    n <- readLn

    print (take n (fib 0 1))
```



```
◑‿◑ a3x3k ◑‿◑ → ./4
Enter the number
6
[0,1,1,2,3,5]
◑‿◑ a3x3k ◑‿◑ → ./4
Enter the number
10
[0,1,1,2,3,5,8,13,21,34]
```

5 - Create a simple calculator with the operations +, -, /, *. Read two numbers and the operation, compute the operation and print the result.

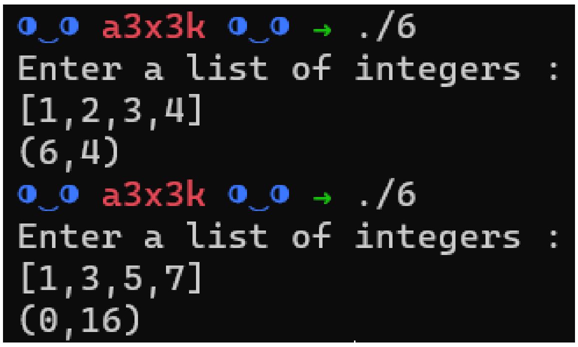
```
sumOf a b c | c == 1 = a + b
 | c == 2 = a - b
 | c == 3 = a * b
 | c == 4 = a / b

main = do
    putStrLn "Enter the Number 1"
    a <- readLn

putStrLn "Enter the Number 2"
    b <- readLn

putStrLn "Enter the 1 for Addition - 2 for Subtraction - 3 for Multiplication -
4 for Division"
    c <- readLn

print (sumOf a b c)
```



```
◐‿◐ a3x3k ◐‿◐ → ./5
Enter the Number 1
45
Enter the Number 2
5
Enter the 1 for Addition - 2 for Subtraction - 3 for Multiplication - 4 for Division
1
50.0
◐‿◐ a3x3k ◐‿◐ → ./5
Enter the Number 1
45
Enter the Number 2
5
Enter the 1 for Addition - 2 for Subtraction - 3 for Multiplication - 4 for Division
2
40.0
◐‿◐ a3x3k ◐‿◐ → ./5
Enter the Number 1
45
Enter the Number 2
5
Enter the 1 for Addition - 2 for Subtraction - 3 for Multiplication - 4 for Division
3
225.0
◐‿◐ a3x3k ◐‿◐ → ./5
Enter the Number 1
45
Enter the Number 2
5
Enter the 1 for Addition - 2 for Subtraction - 3 for Multiplication - 4 for Division
4
9.0
```

6 - Read the list of integers from the user and prints a tuple pair with an even sum and the odd sum of the elements from the list.

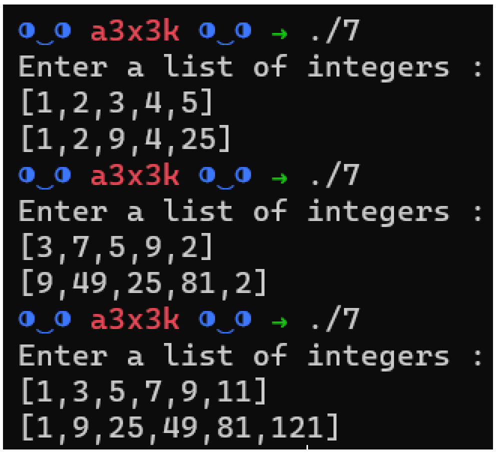
```
find :: Integral a => [a] -> (a,a)

find x = (sum [i | i <- x, i `mod` 2 == 0], sum [i | i <- x, i `mod` 2 /= 0])

main = do

   putStrLn "Enter a list of integers : "

   list <- getLine

   let ls = read list :: [Int]

   print (find ls)
```



```
◑‿◑ a3x3k ◑‿◑ → ./6
Enter a list of integers :
[1,2,3,4]
(6,4)
◑‿◑ a3x3k ◑‿◑ → ./6
Enter a list of integers :
[1,3,5,7]
(0,16)
```

7 - Read a list of integers from the user which prints a list of integers, except that each odd element of the list is replaced by the square of that element.

```
main = do
    putStrLn "Enter a list of integers : "
    list <- getLine
    let ls = read list :: [Int]
    print ([if i `mod` 2 /= 0 then i*i else i | i <- ls])
```



```
◑‿◑ a3x3k ◑‿◑ → ./7
Enter a list of integers :
[1,2,3,4,5]
[1,2,9,4,25]
◑‿◑ a3x3k ◑‿◑ → ./7
Enter a list of integers :
[3,7,5,9,2]
[9,49,25,81,2]
◑‿◑ a3x3k ◑‿◑ → ./7
Enter a list of integers :
[1,3,5,7,9,11]
[1,9,25,49,81,121]
```

Thankyou!!