Higher Order Functions

Code 1:

```
reapply :: (a-> a) -> a -> a
reapply f x = f(f x)

sqr x = x * x
double x = x + x
half x = x/2

main = do
        putStrLn("Higher order functions Double, Half, Square")
        putStr("Double applied twice on 7 : ")
        print( reapply double 7 )
        putStr("Half applied twice on 7 : ")
        print( reapply half 7 )
        putStr("Square applied twice on 7 : ")
        print( reapply sqr 7 )
```

Output 1:

```
Higher order functions Double, Half, Square
Double applied twice on 7 : 28
Half applied twice on 7 : 1.75
Square applied twice on 7 : 2401
```

****

Code 2:

```
mapf :: (a->b) -> [a] -> [b]
mapf f xs = [f x | x <- xs]

main = do
        print( mapf (+1) [1,3,5,7] )
        print( mapf (*2) [1,3,5,7] )
```

Output 2:

```
[2,4,6,8]
[2,6,10,14]
```

****

Code 3:

```
mapf :: (a->b) -> [a] -> [b]

mapf f [] = []
mapf f (x:xs) = f x : mapf f xs

main = do
        print( mapf (+1) [1,3,5,7] )
        print( mapf (*3) [1,3,5,7] )
```

Output 3:

```
[2,4,6,8]
[3,9,15,21]
```

****

Code 4:

```
main = do
        putStr("The division 7/0 is : ")
        print(7/0)
```

Output 4:

```
The division 7/0 is : Infinity
```

****

Code 5:

```
ordpr :: [a] -> [b] -> [(a,b)]

ordpr xs ys = do
                x <- xs
                y <- ys
                return (x,y)

main = print(ordpr [1,2,3] [4,5])
```

Output 5

```
[(1,4),(1,5),(2,4),(2,5),(3,4),(3,5)]
```