```
#Multiplication
import tensorflow as tf
A1 = tf.constant([1, 2, 3, 4])
B1 = tf.constant([3, 4, 5, 5])
C1 = tf.multiply(A1, B1)

print(C1)
```

```
     tf.Tensor([ 3  8 15 20], shape=(4,), dtype=int32)
```

```
#Addition
A1 = tf.constant([1, 2, 3, 4])
B1 = tf.constant([3, 4, 5, 5])
C1 = tf.add(A1, B1)

print(C1)
```

```
     tf.Tensor([4 6 8 9], shape=(4,), dtype=int32)
```

```
#Subtraction
A1 = tf.constant([1, 2, 3, 4])
B1 = tf.constant([3, 4, 5, 5])
C1 = tf.subtract(A1, B1)

print(C1)
```

```
     tf.Tensor([-2 -2 -2 -1], shape=(4,), dtype=int32)
```

```
#MatMul
A1 = tf.constant([[2, 24], [2, 26], [2, 57]])
B1 = tf.constant([[1000], [150]])
C1 = tf.matmul(A1, B1)

print(C1)
```

```
     tf.Tensor(
     [[ 5600]
      [ 5900]
      [10550]], shape=(3, 1), dtype=int32)
```

```
#Reduce_Sum
A1 = tf.Variable([[1,2,3],[3,2,1], [3,3,3]])
B2 = tf.reduce_sum(A1)



B3 = tf.reduce_sum(A1, 0)



B4 = tf.reduce_sum(A1, 1)

print(B2)
print(B3)
print(B4)
```

```
tf.Tensor(21, shape=(), dtype=int32)
tf.Tensor([7 7 7], shape=(3,), dtype=int32)
tf.Tensor([6 6 9], shape=(3,), dtype=int32)
```

```
import tensorflow.compat.v1 as tf


with tf.compat.v1.Session() as sess:
  c = tf.constant([[1, 2], [4, 5]])
  d = tf.constant([[7, 8], [1, 2]])
  res = tf.matmul(c,d)
  print(sess.run(res))
  sess.close()
```

```
[[ 9 12]
 [33 42]]
```

```
with tf.compat.v1.Session() as sess:
  A1 = tf.constant([[2, 24], [2, 26], [2, 57]])
  B1 = tf.constant([[1000], [150]])
  C1 = tf.matmul(A1, B1)
  print(sess.run(C1))
  sess.close()
```

```
[[ 5600]
 [ 5900]
 [10550]]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([1, 2, 4, 5])
  d = tf.constant([7, 8, 1, 2])
  res = tf.add(c,d)
  print(sess.run(res))
  sess.close()
```

```
[ 8 10  5  7]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([1, 2, 4, 5])
  d = tf.constant([7, 8, 1, 2])
  res = tf.multiply(c,d)
  print(sess.run(res))
  sess.close()
```

```
[ 7 16  4 10]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([1, 2, 4, 5])
  d = tf.constant([7, 8, 1, 2])
  res = tf.subtract(c,d)
  print(sess.run(res))
  sess.close()
```

```
    [-6 -6  3  3]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([1, 2, 4, 5])
  res = tf.reduce_sum(c)
  print(sess.run(res))
  sess.close()
```

```
    12
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([[1, 2], [4, 5]])
  res = tf.matrix_transpose(c)
  print(sess.run(res))
  sess.close()
```

```
    [[1 4]
     [2 5]]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([[1.0, 2.0], [4.0, 5.0]])
  res = tf.matrix_determinant(c,name=None)
  print(sess.run(res))
  sess.close()
```

```
    -3.0
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([[1.0, 2.0], [4.0, 5.0]])
  res = tf.matrix_inverse(c)
  print(sess.run(res))
  sess.close()
```

```
    [[-1.6666667   0.6666667 ]
     [ 1.3333334  -0.33333334]]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([[1.0, 2.0], [4.0, 5.0]])
  res = tf.matrix_square_root(c,name=None)
  print(sess.run(res))
  sess.close()
```

```
    [[nan nan]
     [nan nan]]
```

```
with tf.compat.v1.Session() as sess:
  c = tf.constant([[1, 2], [4, 5]])
  d = tf.constant([[7, 8], [1, 2]])
  res =tf.math.minimum(c,d)
  print(sess.run(res))
  sess.close()
```

```
[[1 2]
 [1 2]]
```

```
import tensorflow.compat.v1 as tf


#Basic Operations
with tf.compat.v1.Session() as sess:
  a=tf.constant([1,2,3])
  b=tf.constant([4,7,9])

  sum = tf.add(a, b)

  diff = tf.subtract(a, b)

  prod = tf.multiply(a, b)

  quot = tf.divide(a, b)

  print(sess.run(sum))
  print(sess.run(diff))
  print(sess.run(prod))
  print(sess.run(quot))

  print(sess.run(tf.mod(a,b)))

  print(sess.run(tf.abs(a)))

  print(sess.run(tf.negative(b)))

  print(sess.run(tf.sign(a)))

  print(sess.run(tf.add(1, 2)))

  print(sess.run(tf.square(8)))

  print(sess.run(tf.pow(2,3)))

  print(sess.run(tf.sqrt(4.0)))

  print(sess.run(tf.exp(1.0)))

  print(sess.run(tf.square(4) + tf.square(3)))

  sess.close()
```

```
[ 5  9 12]
[-3 -5 -6]
[ 4 14 27]
[0.25       0.28571429 0.33333333]
[1 2 3]
[1 2 3]
[-4 -7 -9]
```

```
[1 1 1]
3
64
8
2.0
2.7182817
25
```

```
#factorial
with tf.compat.v1.Session() as sess:
 print(sess.run(tf.exp(tf.lgamma(8.0))))
```

```
5040.002
```