```
#1
#Create a lambda function that adds 15 to a given number passed in as an argument
'''
r = lambda a : a + 15
print(r(10))
'''

#*************************************************************************************
*****************************
#2
# lambda expression to return square and cube of a number

#sq_cube = lambda num: (num ** 2, num ** 3) #Return type is tuple
#print(sq_cube(4))

#O/p - (16, 64)


#*************************************************************************************
*****************************
#3.
#Create a lambda function that multiplies 2 arguments
#multiply = lambda a, b : a * b
#print(multiply(5, 2))

#O/p - 10

#*************************************************************************************
*****************************
#4.
```

```
#Program to add 2 numbers

#add = lambda a, b : a + b

#print(add(2, 5))


#O/p - 7

#*****************************************************************************************
****************************

#5.

#Program to solve this expression  a ** 2 + b  ** 2 +  2 *a * b


#expression = lambda a, b : a ** 2 + b  ** 2 +  2 *a * b

#print(expression(5, 3))


#O/p - 64


#*****************************************************************************************
****************************

#6.

#Program to solve this expression 2*a + 3*b + 4*c

#expression = lambda a, b, c : 2 * a + 3 * b + 4 * c

#print(expression(1, 2, 3))


#O/p - 20

#*****************************************************************************************
****************************

#7.

#Program to return a last element of the any sequence

#last_ = lambda sequence: sequence[-1]

#print(last_("hello"))

#O/p - o
```

```
#*********************************************************************************************************
```

#8.

lambda expression to check if the given number is even or odd

#even_odd = lambda num: f"{num} is even" if num % 2 == 0 else f"{num} is odd"

#print(even_odd(4))

#4 is even

```
#*********************************************************************************************************
```

#9.

lambda expression to check if the given string is palindrome or not

#palindrome = lambda string: string == string[::-1]

#print(palindrome("mom"))

#True