```
# #Read two floating point numbers and print the quotient
# # (quantity produced by the division of two numbers).
# n1 = float(input("Enter the first number"))
# n2 = float(input("Enter the second number"))
# print (n1/n2)

# # Read an integer value, which is the duration in seconds of a
certain event,
# # and inform it expressed in hours:minutes:seconds.
# total_seconds = float(input("Enter the time in seconds"))
# hours = total_seconds // 3600
# minutes = total_seconds % 3600
# seconds = minutes % 60
# minutes = minutes // 60
# print("{}:{}:{}".format(int(hours), int(minutes), int(seconds)))

# #Write a function that receives 3 integer values A, B and C as
parameters
# # and returns True if either the sum of the A plus B is lower than
C or C is even,
# # and False otherwise.
# def true_or_false(A, B, C):
#     return A + B < C or C % 2 == 0

# # Write a function that receives two integer numbers (in any
order)
# # and returns the average of all integers between them (both
included).
## solution I
# def average(i1, i2):
#     if i1 > i2:
#         i1, i2 = i2, i1
#     total = 0
#     for i in range(i1, i2+1):
#         total += i
#     return total/(i2+1-i1)
## solution II
# def average(i1, i2):
#     return (i2 + i1)/2

# # Write a function that receives a list of numbers and computes
their product.
# def product(L):
#     prod = 1
#     for n in L:
#         prod = prod * n
#     return prod

# # Write a function that computes and returns the smaller (minimum)
number in a list.
# def minimum(L):
#     #precondition: the list is not empty
#     min = L[0]
#     for elem in L[1:]:
```

```
#         if min > elem:
#             min = elem
#     return min

# # Write a function that receives a list of numbers and a number as
parameters
# # and returns the index where the number appears in the list, or
-1 if the number
# # is not in the list.
# def search (L, x):
#     for i in range(len(L)):
#         if L[i] == x:
#             return i
#     return -1

# print(search([1,2,3,4,5], 3)) #2
# print(search([-124, -18, 0], 1.1)) #-1

# # write a function that receives a string, adds the digits in the
string
# # ignoring the letters and returns the result.
# def add_numbers(string):
#     total = 0
#     for ch in string:
#         if ch >= '0' and  ch <='9':
#             total = total + int(ch)
#     return total

# # Write a function that receives two integer numbers (the number
of rows and columns)
# # and prints an empty square of asterisks in the screen with size
rows * columns.
# # solution I
# def draw_square(row, column):
#     print("*" * column)
#     for r in range(row-2):
#         print("*{}*".format(' ' * (column-2)))
#     print("*" * column)
#
# # Solution II
# def print_row(column, ch):
#     print("*", end='')
#     for c in range(1, column - 1):
#         print(ch, end='')
#     print("*")

# def draw_square(row, column):
#     for r in range(row):
#         ch = ' '
#         if r == 0 or r == row - 1:
#             ch = '*'
#         print_row(column, ch)
# draw_square(2, 2)
```