**Slip 1**

```
my_tuple = (1, 2, 3, 4, 5)
mixed_tuple = (1, "hello", 3.14, True)
element_exists = 3 in my_tuple
my_list = [1, 2, 3, 4, 5]
list_to_tuple = tuple(my_list)
sliced_tuple = my_tuple[1:4]

from functools import reduce
numbers = [1, 2, 3, 4, 5]
product = reduce(lambda x, y: x * y, numbers)
average = reduce(lambda x, y: x + y, numbers) / len(numbers)

import pandas as pd
df = pd.read_csv('Data.csv')
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df['Age'].fillna(df['Age'].mean(), inplace=True)
df = pd.get_dummies(df, columns=['Country'])
```

**Slip 2**

```
n = 5
squares_dict = {x: x*x for x in range(1, n+1)}
sorted_dict = dict(sorted(squares_dict.items()))
max_val = max(squares_dict.values())
min_val = min(squares_dict.values())
sum_val = sum(squares_dict.values())
product_val = reduce(lambda x, y: x * y, squares_dict.values())

numbers = [1, 2, 3, 4, 5]
squared = list(map(lambda x: x*x, numbers))
sum_squares = reduce(lambda x, y: x + y, squared)
tuples_list = [('a', 3), ('b', 1), ('c', 2)]
sorted_tuples = sorted(tuples_list, key=lambda x: x[1])
```

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer, Binarizer
df = pd.read_csv('winequality-red.csv', sep=';')
scaler = MinMaxScaler()
normalized = scaler.fit_transform(df)
standardized = StandardScaler().fit_transform(df)
normalized_data = Normalizer().fit_transform(df)
binarized = Binarizer(threshold=0.5).fit_transform(df)
```

**Slip 3**

```
my_list = [1, 2, 3]
try:
    my_list.nonexistent_method()
except AttributeError:
    print("Attribute error handled")

import re
text = "Contact us at user@example.com or support@test.org"
emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b', text)

import pandas as pd
df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))

def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n-1)

import numpy as np
```

```
diagonal_matrix = np.diag([1, 2, 3])
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])
matrix_product = np.dot(matrix1, matrix2)
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
dot_product = np.dot(array1, array2)

import pandas as pd
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('Data.csv')
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df['Age'].fillna(df['Age'].mean(), inplace=True)
le = LabelEncoder()
df['Purchased'] = le.fit_transform(df['Purchased'])



num = 153
order = len(str(num))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit ** order
    temp //= 10
if num == sum:
    print("Armstrong")
else:
    print("Not Armstrong")

import numpy as np
arr = np.random.randint(0, 100, size=(3, 3))
mean_val = np.mean(arr)
median_val = np.median(arr)
```

```
std_val = np.std(arr)


import pandas as pd
df = pd.read_csv('iris.csv')
print(df.head(10))
print(df.describe())
print(df.dtypes)
df.columns = df.columns.str.replace(' ', '_')
print(df[['sepal_length', 'sepal_width']])
df.to_csv('iris_modified.csv', index=False)



num = 29
if num > 1:
   for i in range(2, int(num**0.5) + 1):
      if num % i == 0:
         print("Not Prime")
         break
   else:
      print("Prime")
else:
   print("Not Prime")

def is_prime(n):
   if n <= 1:
      return False
   for i in range(2, int(n**0.5) + 1):
      if n % i == 0:
         return False
   return True

for num in range(3, 1000, 2):
   if is_prime(num) and is_prime(num + 2):
      print(num, num + 2)
```

```
import pandas as pd
import matplotlib.pyplot as plt
data = {
   'month_number': range(1, 9),
   'face_cream': [2500, 2630, 2140, 3400, 3600, 2760, 2980, 3700],
   'facewash': [1500, 1200, 1340, 1130, 1740, 1555, 1120, 1400],
   'toothpaste': [5200, 5100, 4550, 5870, 4560, 4890, 4780, 5860],
   'bathing_soap': [9200, 6100, 9550, 8870, 7760, 7490, 8980, 9960],
   'shampoo': [1200, 2100, 3550, 1870, 1560, 1890, 1780, 2860],
   'moisturizer': [1500, 1200, 1340, 1130, 1740, 1555, 1120, 1400]
}
df = pd.DataFrame(data)
total_sales = df[['face_cream', 'facewash', 'toothpaste', 'bathing_soap', 'shampoo', 'moisturizer']].sum()
plt.pie(total_sales, labels=total_sales.index, autopct='%1.1f%%')
plt.show()
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.plot(df['month_number'], df['bathing_soap'])
ax2.plot(df['month_number'], df['facewash'])
plt.show()
plt.stackplot(df['month_number'], df['face_cream'], df['facewash'], df['toothpaste'],
          df['bathing_soap'], df['shampoo'], df['moisturizer'])
plt.show()



num = 28
sum = 0
for i in range(1, num):
   if num % i == 0:
      sum += i
if sum == num:
   print("Perfect")
else:
   print("Not Perfect")
```

```
celsius_temps = [0, 20, 30, 40]
fahrenheit = list(map(lambda c: (c * 9/5) + 32, celsius_temps))
strings = ["hello", "world", "python"]
lengths = list(map(lambda s: len(s), strings))

import pandas as pd
df = pd.read_csv('iris.csv')
filtered = df[df['sepal_length'] > 5.0].sort_values('petal_length')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
df['species'] = df['species'].astype('category')
df.to_csv('iris_modified.csv', index=False)



keys = ['a', 'b', 'c']
values = [1, 2, 3]
mapped_dict = dict(zip(keys, values))
tuples_list = [('a', 1), ('b', 2), ('c', 3)]
tuple_to_dict = dict(tuples_list)
original_list = [1, 2, 3, 4, 5]
cloned_list = original_list.copy()
words = ['apple', 'banana', 'cherry', 'date']
long_words = [word for word in words if len(word) > 4]
my_dict = {'a': 1, 'b': 2, 'c': 1, 'd': 3}
unique_dict = {}
for key, value in my_dict.items():
    if value not in unique_dict.values():
        unique_dict[key] = value

import numpy as np
```

```
diagonal_matrix = np.diag([1, 2, 3])
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])
matrix_product = np.dot(matrix1, matrix2)
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
dot_product = np.dot(array1, array2)

import pandas as pd
df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))

import re
text = "Contact us at user@example.com or support@test.org"
emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b', text)

import pandas as pd
df = pd.read_csv('iris.csv')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
df['species'] = df['species'].astype('category')
df.to_csv('iris_modified.csv', index=False)

df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
```

```
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))


keyword = "python"
word_count = {}
with open('sample.txt', 'r') as file:
    for line_num, line in enumerate(file, 1):
        if keyword in line:
            print(f"Line {line_num}: {line.strip()}")
        words = line.split()
        for word in words:
            word_count[word] = word_count.get(word, 0) + 1


import re
text = "Call (123) 456-7890 or 123-456-7890"
phones = re.findall(r'\(\d{3}\)\s\d{3}-\d{4}|\d{3}-\d{3}-\d{4}', text)


try:
    with open('nonexistent.txt', 'r') as file:
        content = file.read()
except FileNotFoundError:
    print("File not found")



add_numbers = lambda x, y: x + y
check_even_odd = lambda x: "Even" if x % 2 == 0 else "Odd"


import pandas as pd
df = pd.read_csv('iris.csv')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
```

```
df['species'] = df['species'].astype('category')
df.to_csv('iris_modified.csv', index=False)

df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))

my_tuple = (1, 2, 3, 4, 5)
mixed_tuple = (1, "hello", 3.14, True)
element_exists = 3 in my_tuple
my_list = [1, 2, 3, 4, 5]
list_to_tuple = tuple(my_list)
sliced_tuple = my_tuple[1:4]

import shutil
shutil.copytree('source_dir', 'backup_dir')

with open('file.c', 'r') as file:
    lines = file.readlines()
with open('file_no_comments.c', 'w') as file:
    for line in lines:
        if not line.strip().startswith('//') and not line.strip().startswith('/*'):
            file.write(line)

import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
df = pd.read_csv('winequality-red.csv', sep=';')
scaler = MinMaxScaler()
normalized = scaler.fit_transform(df)
standardized = StandardScaler().fit_transform(df)
normalized_data = Normalizer().fit_transform(df)
```

```
import numpy as np
arr = np.random.rand(10)
sorted_arr = np.sort(arr)
max_index = np.argmax(sorted_arr)

try:
    with open('file.txt', 'r', encoding='utf-8') as file:
        content = file.read()
except UnicodeDecodeError:
    print("Encoding error")

import pandas as pd
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('Data.csv')
print(df.describe())
print(df.shape)
print(df.head(3))
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df['Age'].fillna(df['Age'].mean(), inplace=True)
df_drop_cols = df.dropna(axis=1)
df_drop_rows = df.dropna()
df_manual = df.fillna(0)
df_ffill = df.fillna(method='ffill')
df_bfill = df.fillna(method='bfill')
df = pd.get_dummies(df, columns=['Country'])

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

```
for num in range(3, 1000, 2):
    if is_prime(num) and is_prime(num + 2):
        print(num, num + 2)


import re
text = "IPs: 192.168.1.1 and 2001:0db8:85a3:0000:0000:8a2e:0370:7334"
ipv4_pattern = r'\b(?:\d{1,3}\.){3}\d{1,3}\b'
ipv6_pattern = r'\b(?:[A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}\b'
ips = re.findall(ipv4_pattern, text) + re.findall(ipv6_pattern, text)

import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
df = pd.read_csv('winequality-red.csv', sep=';')
scaler = MinMaxScaler()
normalized = scaler.fit_transform(df)
standardized = StandardScaler().fit_transform(df)
normalized_data = Normalizer().fit_transform(df)



import numpy as np
arr = np.random.rand(10)
sorted_arr = np.sort(arr)
max_index = np.argmax(sorted_arr)

try:
    with open('file.txt', 'r', encoding='utf-8') as file:
        content = file.read()
except UnicodeDecodeError:
    print("Encoding error")

import pandas as pd
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('Data.csv')
```

```
print(df.describe())
print(df.shape)
print(df.head(3))
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df['Age'].fillna(df['Age'].mean(), inplace=True)
df_drop_cols = df.dropna(axis=1)
df_drop_rows = df.dropna()
df_manual = df.fillna(0)
df_ffill = df.fillna(method='ffill')
df_bfill = df.fillna(method='bfill')
le = LabelEncoder()
df['Purchased'] = le.fit_transform(df['Purchased'])



set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
difference = set1 - set2
is_subset = set1.issubset(set2)
intersection = set1 & set2
union = set1 | set2

numbers = [1, -2, 3, -4, 5, -6]
positive = list(filter(lambda x: x >= 0, numbers))
names = ['Alice', 'Bob', 'Charlie', 'David']
long_names = list(filter(lambda name: len(name) > 5, names))

import pandas as pd
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('Data.csv')
print(df.describe())
print(df.shape)
print(df.head(3))
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df['Age'].fillna(df['Age'].mean(), inplace=True)
```

```
df_drop_cols = df.dropna(axis=1)
df_drop_rows = df.dropna()
df_manual = df.fillna(0)
le = LabelEncoder()
df['Purchased'] = le.fit_transform(df['Purchased'])

keyword = "python"
word_count = {}
with open('sample.txt', 'r') as file:
    for line_num, line in enumerate(file, 1):
        if keyword in line:
            print(f"Line {line_num}: {line.strip()}")
        words = line.split()
        for word in words:
            word_count[word] = word_count.get(word, 0) + 1

my_list = [1, 2, 3]
try:
    my_list.nonexistent_method()
except AttributeError:
    print("Attribute error handled")

from functools import reduce
numbers = [1, 2, 3, 4, 5]
squared = list(map(lambda x: x*x, numbers))
sum_squares = reduce(lambda x, y: x + y, squared)
tuples_list = [('a', 3), ('b', 1), ('c', 2)]
sorted_tuples = sorted(tuples_list, key=lambda x: x[1])

import re
text = "Contact us at user@example.com or support@test.org"
emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b', text)

import pandas as pd
```

```
df = pd.read_csv('iris.csv')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
df['species'] = df['species'].astype('category')
df.to_csv('iris_modified.csv', index=False)

df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))

num = 121
if str(num) == str(num)[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")

from functools import reduce
numbers = [1, 2, 3, 4, 5]
product = reduce(lambda x, y: x * y, numbers)
average = reduce(lambda x, y: x + y, numbers) / len(numbers)

import pandas as pd
import matplotlib.pyplot as plt
data = {
    'month_number': range(1, 11),
    'face_cream': [2500, 2630, 2140, 3400, 3600, 2760, 2980, 3700, 3540, 1990],
    'facewash': [1500, 1200, 1340, 1130, 1740, 1555, 1120, 1400, 1780, 1890],
    'toothpaste': [5200, 5100, 4550, 5870, 4560, 4890, 4780, 5860, 6100, 8300],
```

```
    'bathing_soap': [9200, 6100, 9550, 8870, 7760, 7490, 8980, 9960, 8100, 10300],
    'shampoo': [1200, 2100, 3550, 1870, 1560, 1890, 1780, 2860, 2100, 2300],
    'moisturizer': [1500, 1200, 1340, 1130, 1740, 1555, 1120, 1400, 1780, 1890]
}
df = pd.DataFrame(data)
total_sales = df[['face_cream', 'facewash', 'toothpaste', 'bathing_soap', 'shampoo', 'moisturizer']].sum()
plt.pie(total_sales, labels=total_sales.index, autopct='%1.1f%%')
plt.show()
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.plot(df['month_number'], df['bathing_soap'])
ax2.plot(df['month_number'], df['facewash'])
plt.show()
plt.stackplot(df['month_number'], df['face_cream'], df['facewash'], df['toothpaste'],
              df['bathing_soap'], df['shampoo'], df['moisturizer'])
plt.show()



my_dict = {'a': 1, 'b': 2}
my_dict['c'] = 3
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
merged_dict = {**dict1, **dict2}
for key, value in my_dict.items():
    print(key, value)
key_exists = 'a' in my_dict
del my_dict['a']

import numpy as np
diagonal_matrix = np.diag([1, 2, 3])
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])
matrix_product = np.dot(matrix1, matrix2)
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
```

```
dot_product = np.dot(array1, array2)


import pandas as pd
df = pd.read_csv('Data.csv')
print(df.to_string())



n = 5
squares_dict = {x: x*x for x in range(1, n+1)}
sorted_dict = dict(sorted(squares_dict.items()))
max_val = max(squares_dict.values())
min_val = min(squares_dict.values())
sum_val = sum(squares_dict.values())
product_val = reduce(lambda x, y: x * y, squares_dict.values())

import numpy as np
arr = np.random.randint(0, 100, size=(3, 3))
mean_val = np.mean(arr)
median_val = np.median(arr)
std_val = np.std(arr)

import pandas as pd
df = pd.read_csv('iris.csv')
print(df.head(10))
print(df.describe())
print(df.dtypes)
df.columns = df.columns.str.replace(' ', '_')
print(df[['sepal_length', 'sepal_width']])
df.to_csv('iris_modified.csv', index=False)



with open('sample.txt', 'r') as file:
    content = file.read()
    words = len(content.split())
```

```
    lines = len(content.splitlines())
    chars = len(content)
    print(f"Words: {words}, Lines: {lines}, Chars: {chars}")

import re
text = "This  has   multiple   spaces"
cleaned = re.sub(r'\s+', ' ', text)

import pandas as pd
df = pd.read_csv('Student_bucketing.csv')
print(df.head())
bins = [0, 20, 40, 60, 80, 100]
labels = ['Poor', 'Below_average', 'Average', 'Above_average', 'Excellent']
df['marks_bucket'] = pd.cut(df['marks'], bins=bins, labels=labels)
print(df.head(10))



rows = 4
for i in range(1, rows + 1):
    for j in range(i, rows + 1):
        print(j, end=" ")
    print()

import pandas as pd
df = pd.read_csv('iris.csv')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
df['species'] = df['species'].astype('category')
df.columns = df.columns.str.replace(' ', '_')
print(df[['sepal_length', 'sepal_width']])
```

```
import numpy as np
arr1 = np.random.rand(5)
arr2 = np.random.rand(5)
add = arr1 + arr2
sub = arr1 - arr2
mul = arr1 * arr2
div = arr1 / arr2



rows = 4
for i in range(1, rows + 1):
    for j in range(65, 65 + i):
        print(chr(j), end=" ")
    print()

set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
symmetric_diff = set1 ^ set2
value_present = 3 in set1
no_common = set1.isdisjoint(set2)
set1 -= set2

import requests
from bs4 import BeautifulSoup
response = requests.get('http://example.com')
soup = BeautifulSoup(response.text, 'html.parser')
urls = [a['href'] for a in soup.find_all('a', href=True)]
with open('urls.txt', 'w') as file:
    for url in urls:
        file.write(url + '\n')

import numpy as np
arr_3d = np.random.rand(2, 3, 4)
flattened = arr_3d.flatten()
```

```
import pandas as pd
data = {'RN': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Charlie'], 'Percentage': [85.5, 92.0, 78.5]}
df = pd.DataFrame(data)
print(df.head())
print(df.describe())
print(df.dtypes)

import shutil
shutil.copytree('source_dir', 'backup_dir')

with open('file.c', 'r') as file:
    lines = file.readlines()
with open('file_no_comments.c', 'w') as file:
    for line in lines:
        if not line.strip().startswith('//') and not line.strip().startswith('/*'):
            file.write(line)

even_sum = sum(range(2, 21, 2))
marks = []
for i in range(3):
    marks.append(float(input(f"Enter mark {i+1}: ")))
marks.sort()
best_avg = (marks[1] + marks[2]) / 2

import re
text = "IPs: 192.168.1.1 and 2001:0db8:85a3:0000:0000:8a2e:0370:7334"
ipv4_pattern = r'\b(?:\d{1,3}\.){3}\d{1,3}\b'
ipv6_pattern = r'\b(?:[A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}\b'
ips = re.findall(ipv4_pattern, text) + re.findall(ipv6_pattern, text)

import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
df = pd.read_csv('winequality-red.csv', sep=';')
scaler = MinMaxScaler()
```

```
normalized = scaler.fit_transform(df)
standardized = StandardScaler().fit_transform(df)
normalized_data = Normalizer().fit_transform(df)

int_num = 10
float_num = 3.14
complex_num = 2 + 3j
num = 25
sqrt = num ** 0.5
length = 5
width = 3
area_rect = length * width
side = 4
area_square = side * side
perimeter_square = 4 * side

import pandas as pd
df = pd.read_csv('iris.csv')
grouped = df.groupby('species').mean()
df['sepal_ratio'] = df['sepal_length'] / df['sepal_width']
df = df.drop('sepal_ratio', axis=1)
df = df.dropna()
df['petal_length'] = df['petal_length'].fillna(df['petal_length'].mean())
df['species'] = df['species'].astype('category')
df.columns = df.columns.str.replace(' ', '_')
print(df[['sepal_length', 'sepal_width']])

import numpy as np
arr1 = np.random.rand(5)
arr2 = np.random.rand(5)
add = arr1 + arr2
sub = arr1 - arr2
mul = arr1 * arr2
div = arr1 / arr2
```