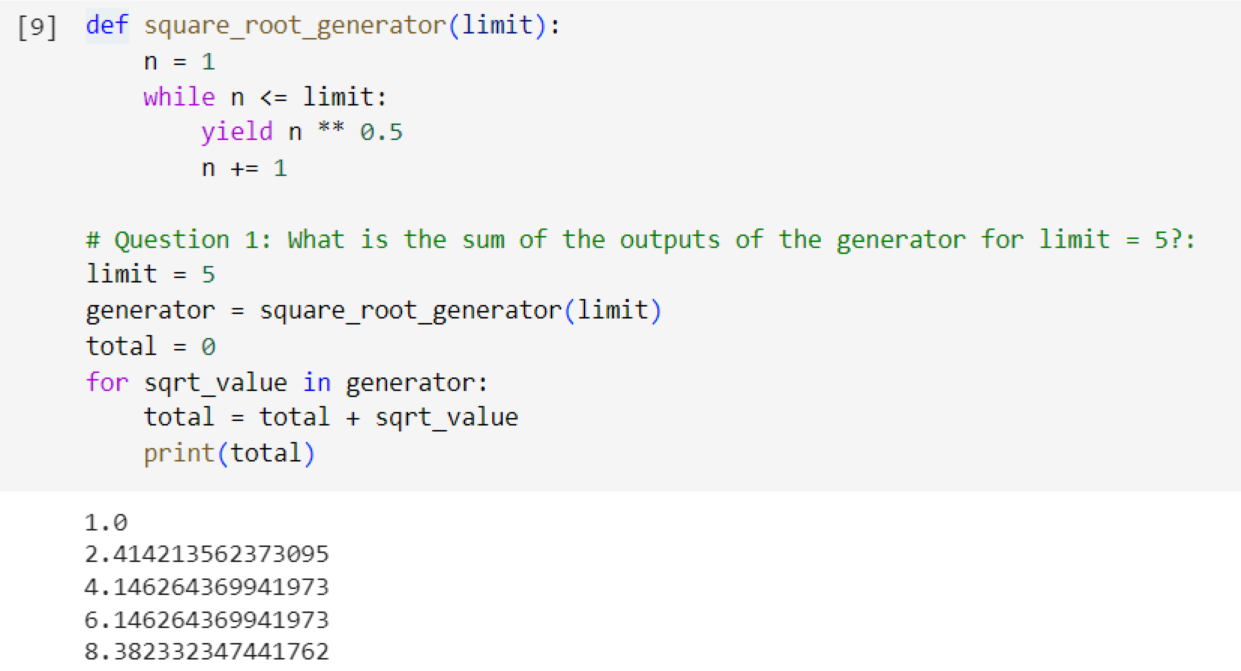
Homework

Question 1: What is the sum of the outputs of the generator for limit = 5?

- A: 10.23433234744176
- B: 7.892332347441762
- **C: 8.382332347441762**
- D: 9.123332347441762

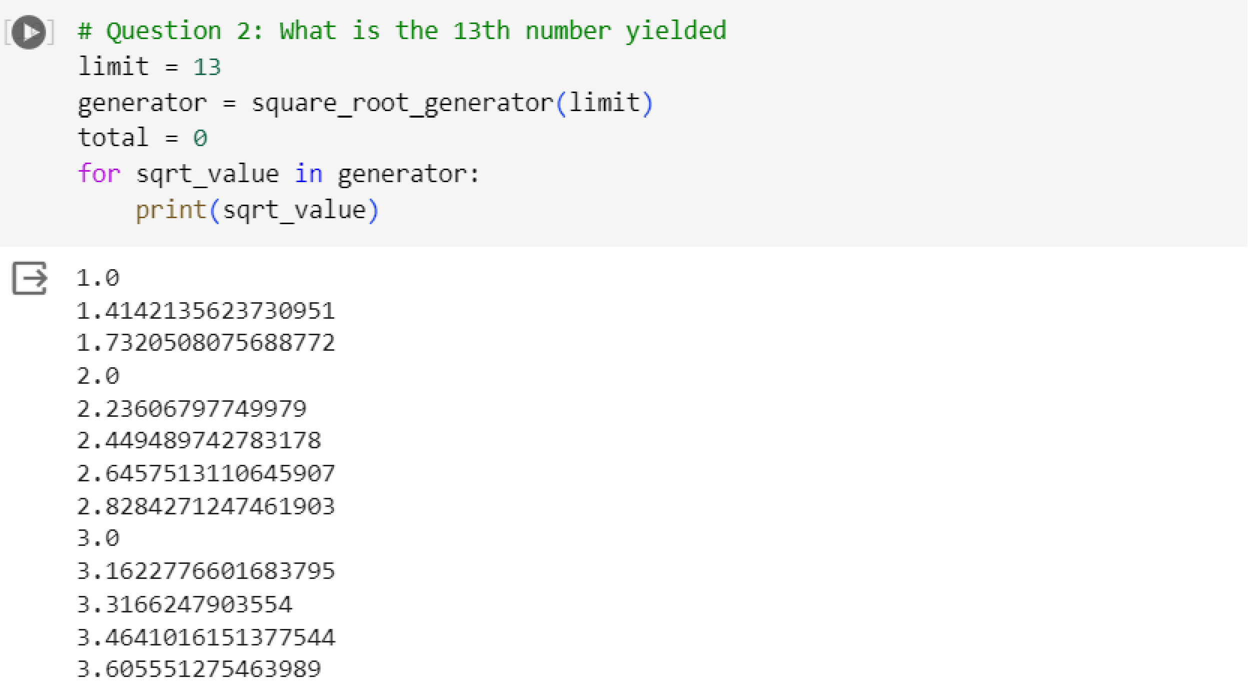


```
[9]  def square_root_generator(limit):
         n = 1
         while n <= limit:
             yield n ** 0.5
             n += 1

     # Question 1: What is the sum of the outputs of the generator for limit = 5?:
     limit = 5
     generator = square_root_generator(limit)
     total = 0
     for sqrt_value in generator:
         total = total + sqrt_value
         print(total)
```

```
1.0
2.414213562373095
4.146264369941973
6.146264369941973
8.382332347441762
```

Question 2: What is the 13th number yielded by the generator?

- A: 4.236551275463989
- **B: 3.605551275463989**
- C: 2.345551275463989
- D: 5.678551275463989

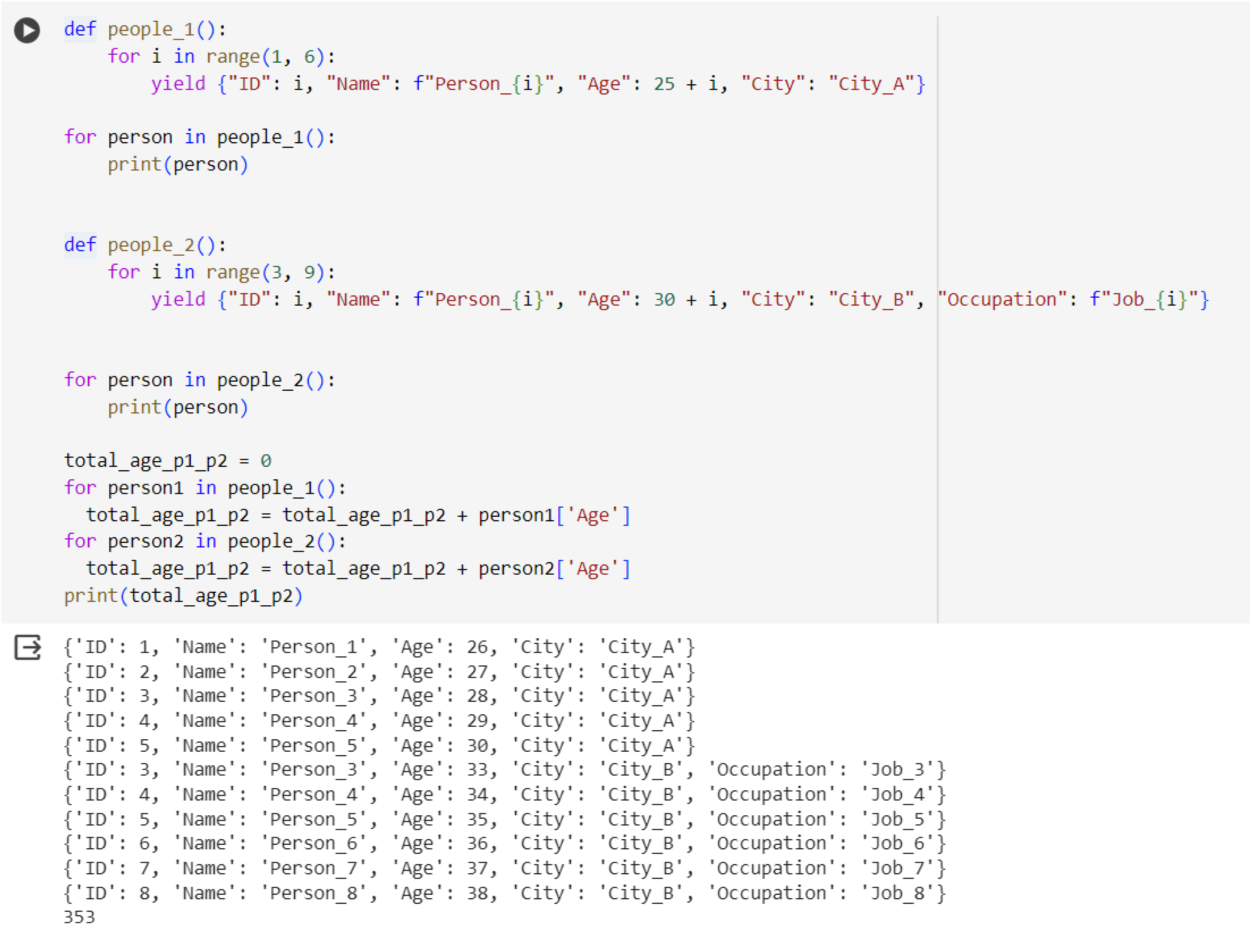


```python
# Question 2: What is the 13th number yielded
limit = 13
generator = square_root_generator(limit)
total = 0
for sqrt_value in generator:
    print(sqrt_value)
```

```
1.0
1.4142135623730951
1.7320508075688772
2.0
2.23606797749979
2.449489742783178
2.6457513110645907
2.8284271247461903
3.0
3.1622776601683795
3.3166247903554
3.4641016151377544
3.605551275463989
```

Question 3: Append the 2 generators. After correctly appending the data, calculate the sum of all ages of people.

- **A: 353**
- B: 365
- C: 378
- D: 390

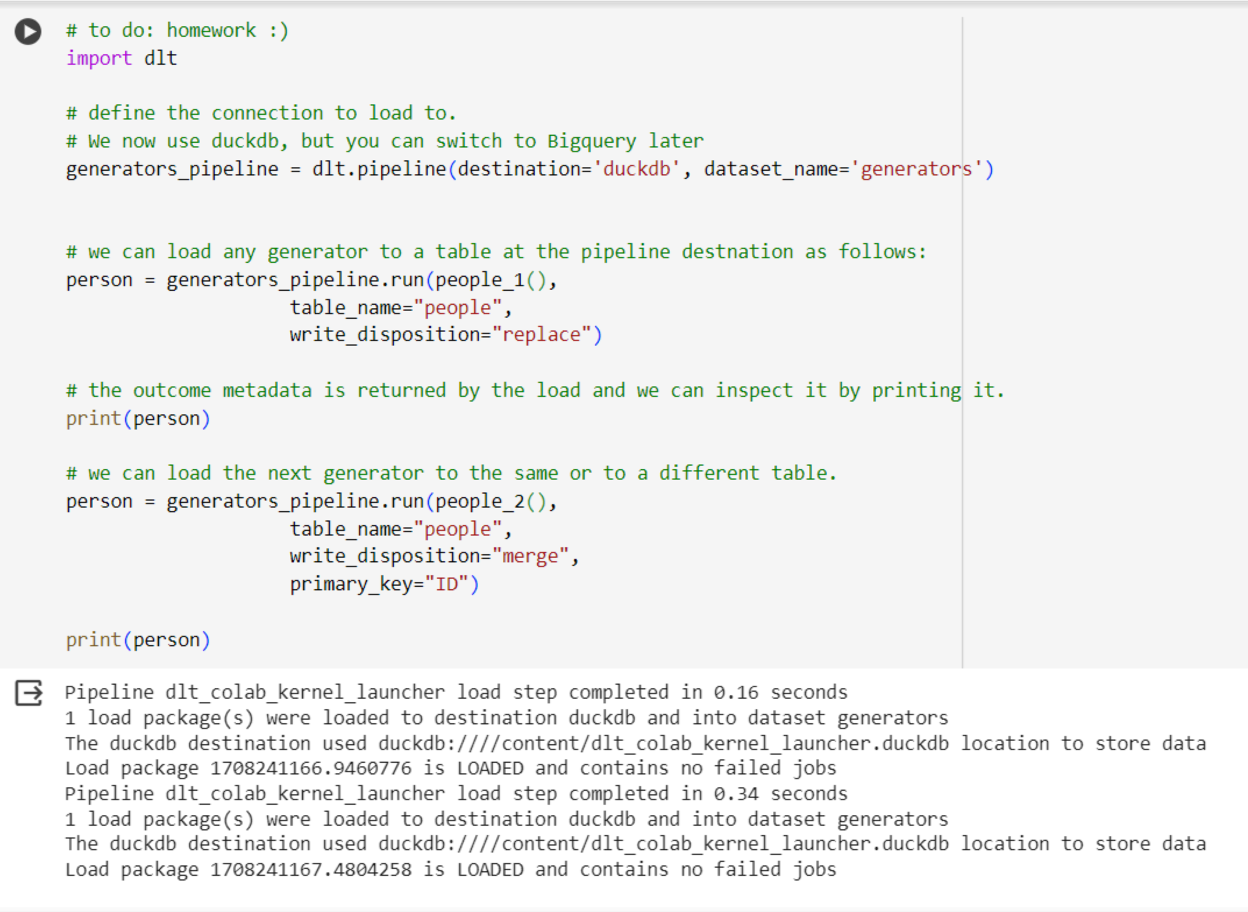


```python
def people_1():
    for i in range(1, 6):
        yield {"ID": i, "Name": f"Person_{i}", "Age": 25 + i, "City": "City_A"}

for person in people_1():
    print(person)


def people_2():
    for i in range(3, 9):
        yield {"ID": i, "Name": f"Person_{i}", "Age": 30 + i, "City": "City_B", "Occupation": f"Job_{i}"}


for person in people_2():
    print(person)

total_age_p1_p2 = 0
for person1 in people_1():
  total_age_p1_p2 = total_age_p1_p2 + person1['Age']
for person2 in people_2():
  total_age_p1_p2 = total_age_p1_p2 + person2['Age']
print(total_age_p1_p2)
```

```
{'ID': 1, 'Name': 'Person_1', 'Age': 26, 'City': 'City_A'}
{'ID': 2, 'Name': 'Person_2', 'Age': 27, 'City': 'City_A'}
{'ID': 3, 'Name': 'Person_3', 'Age': 28, 'City': 'City_A'}
{'ID': 4, 'Name': 'Person_4', 'Age': 29, 'City': 'City_A'}
{'ID': 5, 'Name': 'Person_5', 'Age': 30, 'City': 'City_A'}
{'ID': 3, 'Name': 'Person_3', 'Age': 33, 'City': 'City_B', 'Occupation': 'Job_3'}
{'ID': 4, 'Name': 'Person_4', 'Age': 34, 'City': 'City_B', 'Occupation': 'Job_4'}
{'ID': 5, 'Name': 'Person_5', 'Age': 35, 'City': 'City_B', 'Occupation': 'Job_5'}
{'ID': 6, 'Name': 'Person_6', 'Age': 36, 'City': 'City_B', 'Occupation': 'Job_6'}
{'ID': 7, 'Name': 'Person_7', 'Age': 37, 'City': 'City_B', 'Occupation': 'Job_7'}
{'ID': 8, 'Name': 'Person_8', 'Age': 38, 'City': 'City_B', 'Occupation': 'Job_8'}
353
```

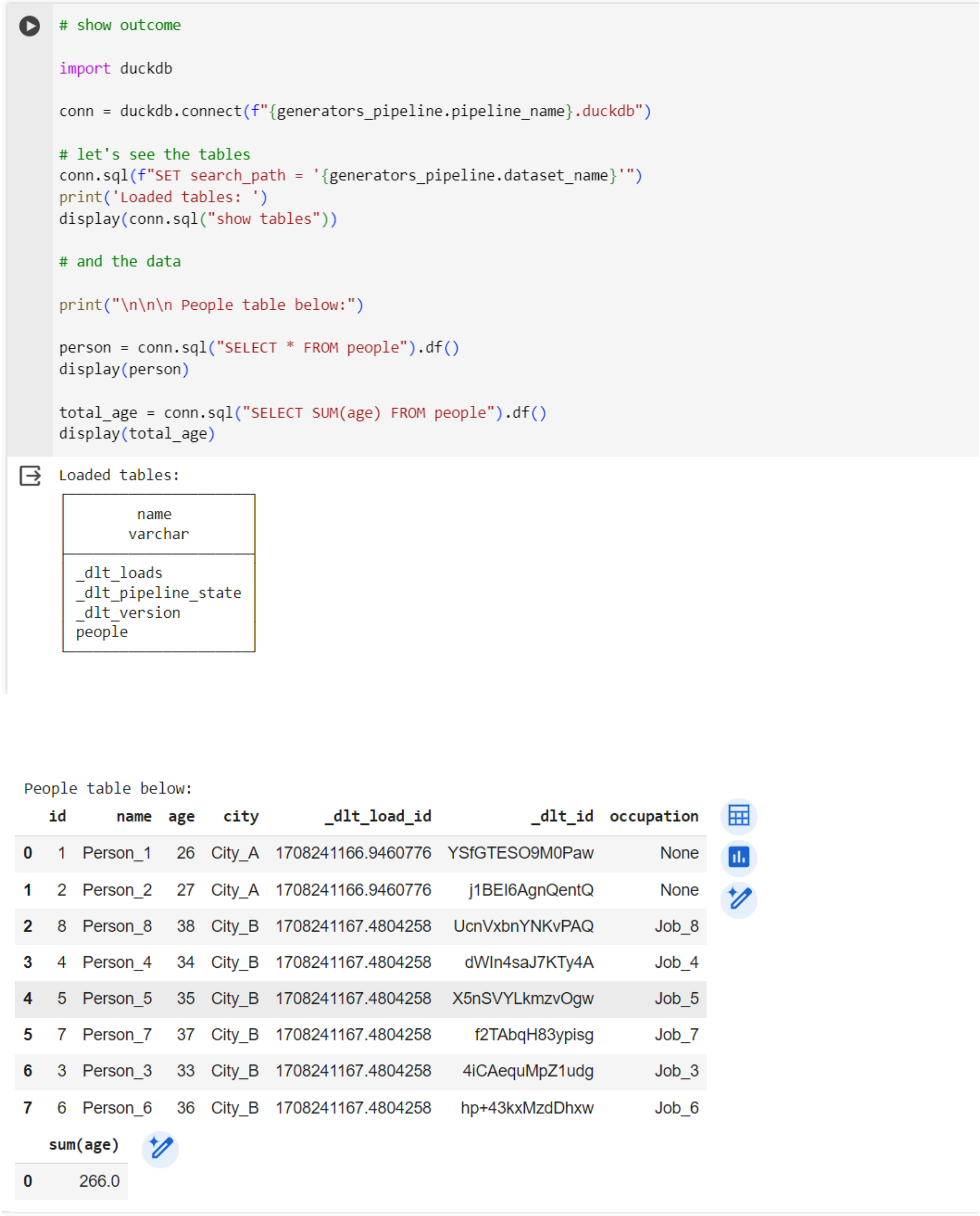
Question 4: Merge the 2 generators using the ID column. Calculate the sum of ages of all the people loaded as described above.

- A: 215
- **B: 266**
- C: 241
- D: 258



```
# to do: homework :)
import dlt

# define the connection to load to.
# We now use duckdb, but you can switch to Bigquery later
generators_pipeline = dlt.pipeline(destination='duckdb', dataset_name='generators')


# we can load any generator to a table at the pipeline destnation as follows:
person = generators_pipeline.run(people_1(),
                    table_name="people",
                    write_disposition="replace")

# the outcome metadata is returned by the load and we can inspect it by printing it.
print(person)

# we can load the next generator to the same or to a different table.
person = generators_pipeline.run(people_2(),
                    table_name="people",
                    write_disposition="merge",
                    primary_key="ID")

print(person)
```

```
Pipeline dlt_colab_kernel_launcher load step completed in 0.16 seconds
1 load package(s) were loaded to destination duckdb and into dataset generators
The duckdb destination used duckdb:////content/dlt_colab_kernel_launcher.duckdb location to store data
Load package 1708241166.9460776 is LOADED and contains no failed jobs
Pipeline dlt_colab_kernel_launcher load step completed in 0.34 seconds
1 load package(s) were loaded to destination duckdb and into dataset generators
The duckdb destination used duckdb:////content/dlt_colab_kernel_launcher.duckdb location to store data
Load package 1708241167.4804258 is LOADED and contains no failed jobs
```



```python
# show outcome

import duckdb

conn = duckdb.connect(f"{generators_pipeline.pipeline_name}.duckdb")

# let's see the tables
conn.sql(f"SET search_path = '{generators_pipeline.dataset_name}'")
print('Loaded tables: ')
display(conn.sql("show tables"))

# and the data

print("\n\n\n People table below:")

person = conn.sql("SELECT * FROM people").df()
display(person)

total_age = conn.sql("SELECT SUM(age) FROM people").df()
display(total_age)
```

```
Loaded tables:
┌─────────────────────┐
│        name         │
│       varchar       │
├─────────────────────┤
│ _dlt_loads          │
│ _dlt_pipeline_state │
│ _dlt_version        │
│ people              │
└─────────────────────┘
```

```
 People table below:
```

	id	name	age	city	_dlt_load_id	_dlt_id	occupation
0	1	Person_1	26	City_A	1708241166.9460776	YSfGTESO9M0Paw	None
1	2	Person_2	27	City_A	1708241166.9460776	j1BEI6AgnQentQ	None
2	8	Person_8	38	City_B	1708241167.4804258	UcnVxbnYNKvPAQ	Job_8
3	4	Person_4	34	City_B	1708241167.4804258	dWIn4saJ7KTy4A	Job_4
4	5	Person_5	35	City_B	1708241167.4804258	X5nSVYLkmzvOgw	Job_5
5	7	Person_7	37	City_B	1708241167.4804258	f2TAbqH83ypisg	Job_7
6	3	Person_3	33	City_B	1708241167.4804258	4iCAequMpZ1udg	Job_3
7	6	Person_6	36	City_B	1708241167.4804258	hp+43kxMzdDhxw	Job_6

	sum(age)
0	266.0