Data Masking:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Employee ID | Name | User ID | Gender | Age | Phone Number | Date of Birth |
| 57861 | Ralph Kendrick | dem55 | O | 35 | 7758956347 | 20-05-1988 |
| 18953 | Dorothy Leadbetter | har83 | M | 20 | 9150357745 | 07-07-2003 |
| 33291 | Greg Rm | kja96 | F | 18 | 7865489007 | 04-10-2005 |
| 96272 | Craig King | jsm77 | F | 44 | 9374674890 | 16-08-1979 |
| 7195 | Linda Bahn | pnx31 | M | 65 | 7689988542 | 29-09-1958 |
| 94757 | Liane Lavin | jdb81 | F | 43 | 9347897140 | 08-07-1980 |
| 48167 | Mary James | bwg34 | O | 57 | 9445678912 | 13-04-1966 |
| 50730 | Kenneth Bass | fsk57 | F | 45 | 9946771966 | 28-02-1978 |
| 46122 | Matthew Lara | rdv50 | M | 28 | 9994034782 | 25-03-1995 |
| 78912 | Greig Smith | grs10 | O | 32 | 9387361103 | 01-01-1991 |

|  |  |  |  |
| --- | --- | --- | --- |
| Driver's License Number | Passport Number | Bank Account Number | Email address |
| US2320217789012 | A3903463 | 15432964088 | [kend634@gmail.com](mailto:kend634@gmail.com) |
| US1420201474654 | B5238914 | 11653894257 | [lead774@gmail.cmm](mailto:lead774@gmail.cmm) |
| US8520228548916 | P8539510 | 12546729086 | [graig634@gmail.com](mailto:graig634@gmail.com) |
| US3420211174590 | A8239058 | 26789259789 | [king489@gmail.com](mailto:king489@gmail.cfm) |
| US2120213382901 | U6452890 | 85372897256 | [behn854@gmail.com](mailto:behn854@gmail.com) |
| US9020226739034 | N6349025 | 67387098734 | [lavi714@gmail.com](mailto:lavi714@gmail.com) |
| US1120194590763 | S7450928 | 52890156238 | [jame891@gmail.com](mailto:jame891@gmail.com) |
| US2720237320165 | P6428904 | 62296086118 | [bass196@gmail.com](mailto:bass196@gmail.com) |
| US3420212890123 | C6328942 | 45396802167 | [lara478@gmail.com](mailto:lara478@gmail.com) |
| US2720196219054 | F7501358 | 16724390568 | [smit110@gmail.com](mailto:smit110@gmail.com) |

|  |  |  |  |
| --- | --- | --- | --- |
| Weight | D.NO | Street name | Pin code |
| 89.74 | c-20 | Park Avenue | 625016 |
| 56.56 | E-85 | Washing Tone Street | 624902 |
| 66.5 | S-60 | Crosby Street | 623917 |
| 94.69 | W-30 | Fifth Avenue Street | 623908 |
| 70.86 | D-9 | Broadway | 536894 |
| 80.78 | F-8 | St Mark's | 456789 |
| 61.65 | G-9 | Madison Avenue | 345678 |
| 70.74 | C-22 | NYC street | 536478 |
| 50.67 | D-59 | Jones Street | 657383 |
| 68.39 | R-20 | Walls Street | 643589 |

\*The above data which we are working on contains all the data types for performing data masking.

**NAME:**

## Problem statement: To mask a list of Names.

## Inputs: A list of Names.

## Outputs: To make a list of fake names (Mask the real one)

## Pseudocode:

1. Generating Second Name:
2. Keep the first name and generate random names for second name.
3. Using mimesis library, generate a fake one for second name.
4. Return the Masked Name.

E.g.: **Input:** Emily Holmes

By using mimesis, we generate random names for second name.

**Output:** Emily George

1. Print by first Character
   * + 1. For all the first characters of the name, we assign specific individual name.
       2. Return the masked name.

E.g.: **Input:** Karthick Jones

For K, we assigned Ralph Haston

**Output:** Ralph Haston

1. Specific expression for any characters:
   * + 1. For specific characters in alphabet, print asterisk (\*)
       2. Return the masked name.

E.g.: **Input:** Ralph Haston

Here R, O=\*

**Output:** \*alph Hast\*n

1. Scattering names:
   * + 1. For all the names in the given list, print another name from list.
       2. Count the characters in first string and print the name in that count.
       3. Return Masked name.

E.g.: **Input:** Enzo Thomas, here Enzo has 4 characters

Then the name in the fourth place gets printed

**Output**: Ralph Haston (which was in fourth place gets printed)

1. Printing by random number:
   * + 1. Create an empty list.
       2. By using random library, the Name in the randomly generated number get printed first.
       3. Return the masked name.

E.g.: **Input:** Emily Chris.

By using random library, we get 10.

Then the name in 10th row gets printed in the first place.

**Output:** Damon Salvatore.

1. Shuffling and generating fake one:
2. For all the names in the odd place, gets shuffled among itself.
3. For all the names in the even place, gets generated using faker.
4. Return the masked data.
5. List of random names:
   * + 1. Create an empty list.
       2. Print the name from the list in a random manner.
       3. Return the masked data.
6. Random characters for the length:
   * + 1. For the length of the name
       2. Using the random library, we can generate random characters for the length of the name.
       3. Return the masked name.

E.g.: **Input:** Emily Thomas.

When Emily met Georgia, she was stunned

**Output:** Shaby Dwqkay.

1. Random characters for the length of first name:
   1. For the length of the first name
   2. Using the random library, we can generate random characters for the length of the first name.
   3. Return the masked name.

E.g.: **Input:** Emily Thomas.

**Output:** Shaby Thomas.

1. Random characters for the length of second name:
   1. For the length of the second name
   2. Using the random library, we can generate random characters for the length of the second name.
   3. Return the masked name.

E.g.: **Input:** Emily Thomas.

**Output:** Emily Dequay

1. Divide and shuffle among name itself:
2. Divide the first and second name.
3. First name shuffle among itself, second name shuffle among itself
4. Print the masked name.
5. Divide and shuffle among name itself:
6. Divide the first and second name.
7. First name shuffle among itself, second name shuffle among itself
8. First names printed as second and second name printed as first.
9. Print the masked name.
10. Divide and shuffle the names:
11. Divide all the names into two list.
12. Shuffle among first list, shuffle among second list.
13. Print the first to second list and vice versa.
14. Print the masked data.
15. Shuffling by odd and even:
16. Shuffling the names.
17. For the names in odd, shuffle among odd.
18. For the names in even, shuffle among even.
19. Print the masked name.
20. Durstenfeld shuffle:
21. It shuffles the list in place and requires only a single pass through the list.
22. Masking second name:
23. Masking the second name:
24. Keep the first name.
25. Mask the second one using special characters.
26. Return the masked name.

E.g.: **Input:** Damon Salvatore.

**Output:** Damon \*\*\*\*\*\*\*\*\*

1. Masking first name:
2. Masking the first name:
3. Keep the second name.
4. Mask the first one using special characters.
5. Return the masked name.

E.g.: **Input:** Damon Salvatore.

**Output:** \*\*\*\*\* Salvatore.

1. Using AND operator:
2. Using and operator on ascii value:
3. For each character, their ascii value get converted to binary number.
4. Using And operator, join the converted binary number and binary of 5.
5. Return the masked data.

E.g.: **Input:** Emily Thomas.

**Output:** 1mily Dwqkay.

1. Using OR operator:
2. Using or operator on ascii value:
3. For each character, their ascii value get converted to binary number.
4. Using Or operator, join the converted binary number and binary of 5.
5. Return the masked data.

E.g.: **Input:** Emily Thomas.

**Output:** 1mily Dwqkay.

1. Masking the whole name:

Text

Description automatically generated

1. Fisher-Yates shuffle algorithm (also known as the Knuth shuffle algorithm):
2. Iterating over the list of items in reverse order, and at each iteration, swapping the current item with a random item from the remaining unshuffled items.

## 5.Code implementation:

**WEIGHT:**

## Problem statement: To mask a list of Given weights.

## Inputs: A list of masked weights.

## Outputs: To make a list of masked weights (Mask the real one)

## Pseudocode:

1. Shuffle the numbers:
2. Print the second two decimals in first place.

Input: 34.89

Output:89.34

1. Shuffle the first digit:
2. Shuffle the first and second integer.

Input: 34.89

Output:43.89

1. Shuffle the first digit in decimal:
2. Shuffle the first and second decimal.

Input: 34.89

Output:34.98

1. Shuffling among given data:
2. In the list of given weights, shuffle among itself.
3. Adding numbers
4. In the given weight, add first two digit and print them in output.

Input:34.89

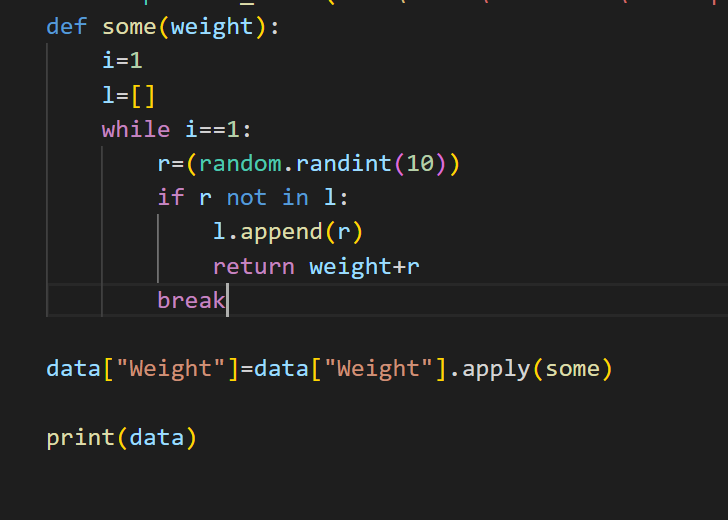
Output:70.89

1. Subtracting numbers using threshold:
2. Subtract the first two numbers and print by threshold values.
3. If the number is between 0-5, print 45+value.
4. If the number is between 6-10, print 55+value.
5. Multiplying numbers:
6. Multiply the first two numbers and print by threshold values.
7. If the number is between1-20, print 45+value.
8. If the number is between 20-50, print 35+value.
9. If the number is between 60-100, print 10+value.
10. Cross printing:
11. In the given list of data, all the numbers in the integer side gets shuffled among itself.
12. Cross printing:
13. In the given list of data, all the numbers in the decimal side gets shuffled among itself.
14. Printing by adding value:
15. All the given list of numbers gets added by random number in the range of 1-10

Input:34.89

34.89+10.4

Output:45.29



1. Printing by subtracting value:
2. All the given list of numbers gets subtracted by random number of 1-10.

Input:34.89

34.89+10.4

Output:24.49

Text

Description automatically generated

1. Shuffling first and last digit:
2. The number in first place get printed in the last place.

Input:34.89

Output:94.83

1. Shuffling second and third digit:
2. The number in second place get printed in the third place.

Input:34.89

Output:38.49

**DATE OF BIRTH:**

## Problem statement: To mask a list of Given DOB.

## Inputs: A list of masked DOB.

## Outputs: To make a list of masked DOB (Mask the real one)

## Pseudocode:

1. Using threshold:

For the year component, keep the threshold between the highest and lowest year, print randomly among it.

Input:2000,2015,2008

Output:2005,2012,2014

1. Using random:

For the month, it generates random month.

Input:05/05/2015

Output:05/09/2015

1. Using random for dates:

For the dates, it generates random dates.

Input:05/05/2015

Output:25/05/2015

1. Shuffling dates:

Among the given dates, we can shuffle the dates.

1. Assigning values:

For specific month, we can assign specific values.

Likewise, it can be built for dates and years.

Input:09/04/2002

For april: 06,14,25.

Output:14/04/2002

1. Adding dates:

Adding the dates, printing the result.

Input:28/07/2013

Output:10/07/2013

1. Using string slicing:

Input: 14/04/2002

Output:14/\*\*/\*\*\*\*

1. Using regular expressions:

Input:05/05/2015

Output: xx/xx/2015

**AGE:**

## Problem statement: To mask a list of Given DOB.

## Inputs: A list of masked DOB.

## Outputs: To make a list of masked DOB (Mask the real one)

## Pseudocode:

1. Using random library:
2. Using Random library, get the number from 1 to 10.
3. Add the random number to the given age.
4. Return Age.

Input:45 [45+5=50]

Output:50

1. Shuffling the age:

From the given list of age, we shuffle the age.

1. Adding the age:

Adding the age, if it goes below threshold then print original age.

1. Subtracting and assigning:

* Subtract the age, if the values fall among the given lists.
* Then print the assigned value.

Input:45

1=[5-4]

l=[35,45,65,75]

Output:65

1. Rounding age to the nearest 5 or 10 years:

* If the unit number is between 0-5, it round the value to 5.
* If the unit number is between 6-10, it round the value to 10.

1. Approximating age based on age groups:

* For the age<30: it prints 30-35
* For the 30<age<40: it prints 40-45
* For the 40<age<50: it prints 50-55…

1. Odd order:

For the age in odd place, get multiplied.

Input:47

Output:28

1. Even order:

For the age in even place, get added.

Input:47

Output:11

1. Using asterisk:

Input:47

Output:\*\*

1. Shuffling the age:

Input:67

Output:76