# Lab – 7 List

Tasks:

1. Draw a diagram to show the memory allocation for each of the program segments. The first one have been done for you.

|  |  |
| --- | --- |
| i) friends = [ "Peter", "Susan", "Mary" ]  draw list of **friends** | "Peter"  "Susan"  "Mary"  0 1 2  friends |
| ii) list1 = [ "a", "b" ]  list2 = [1, 2]  list3 = list1 + list2  draw **list3** | "a"  0 1 2 3  list3  "b"  1  2 |
| iii) numbers = list ( range ( 5, 35, 10) )  draw list of **numbers** | 5  15  25  0 1 2  numbers |
| iv) string = "hello"  list1 = string.split ( )  draw **list1** | "hello"  0  list1 |
| v) string = "hello how are you?"  list2 = string.split ( )  draw **list2** | "hello"  0 1 2 3  list2  "how"  "are"  "you?" |

1. Write a function to loop through and print out all **even numbers** from the numbers list in the same order they are received. Don’t print any numbers that come **on or after 237** in the sequence.

# your code goes here

# Output Generation. You are not allowed to modify the following codes

def main():

numbers = [

951, 402, 984, 651, 360, 69, 408, 319, 601, 485, 980, 507, 725,

547, 544, 615, 83, 165, 141, 501, 263, 617, 865, 575, 219, 390,

984, 592, 236, 105, 942, 941, 386, 462, 47, 418, 907, 344, 236,

375, 823, 566, 597, 978, 328, 615, 953, 345, 399, 162, 758, 219,

918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815,

67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445,

742, 717, 958, 609, 842, 451, 688, 753, 854, 685, 93, 857, 440,

380, 126, 721, 328, 753, 470, 743, 527

]

print\_list\_even(numbers)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

402

984

360

…

918

Answer:

def print\_list\_even(numbers):

for number in numbers:

if number == 237:

break

if number % 2 == 1:

continue

print(number)

1. Write a function to loop through and count all odd, even and prime numbers from the numbers list provided and return the statistics on counted numbers back to the main function:

# your code goes here

# Output Generation. You are not allowed to modify the following codes

def main():

numbers = [

951, 402, 984, 651, 360, 69, 408, 319, 601, 485, 980, 507, 725,

547, 544, 615, 83, 165, 141, 501, 263, 617, 865, 575, 219, 390,

984, 592, 236, 105, 942, 941, 386, 462, 47, 418, 907, 344, 236,

375, 823, 566, 597, 978, 328, 615, 953, 345, 399, 162, 758, 219,

918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815,

67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445,

742, 717, 958, 609, 842, 451, 688, 753, 854, 685, 93, 857, 440,

380, 126, 721, 328, 753, 470, 743, 527

]

odd\_number, even\_number, prime\_number = counting\_on\_list(numbers)

print("Total no. of odd numbers:",odd\_number)

print("Total no. of even numbers:",even\_number)

print("Total no. of prime numbers:",prime\_number)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Total no. of odd numbers: 55

Total no. of even numbers: 45

Total no. of prime numbers: 14

Answer:

# your code goes here

def counting\_on\_list(numbers):

odd\_number = 0

even\_number = 0

prime\_number = 0

for i in numbers:

if i % 2 == 0:

even\_number += 1

else:

odd\_number += 1

for divisor in range(2, i):

if i % divisor == 0:

break

else:

prime\_number += 1

return odd\_number, even\_number, prime\_number

# Output Generation. You are not allowed to modify the following codes

def main():

numbers = [

951, 402, 984, 651, 360, 69, 408, 319, 601, 485, 980, 507, 725,

547, 544, 615, 83, 165, 141, 501, 263, 617, 865, 575, 219, 390,

984, 592, 236, 105, 942, 941, 386, 462, 47, 418, 907, 344, 236,

375, 823, 566, 597, 978, 328, 615, 953, 345, 399, 162, 758, 219,

918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815,

67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445,

742, 717, 958, 609, 842, 451, 688, 753, 854, 685, 93, 857, 440,

380, 126, 721, 328, 753, 470, 743, 527

]

odd\_number, even\_number, prime\_number = counting\_on\_list(numbers)

print("Total no. of odd numbers:",odd\_number)

print("Total no. of even numbers:",even\_number)

print("Total no. of prime numbers:",prime\_number)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Write a program that uses a list to store a set of random numbers and modify the values in the list upon the request of the user.

# your code goes here

...

...

# Output Generation. You are not allowed to modify the following codes

def main():

no\_of\_random\_number = int(input("How many random numbers?"))

min\_random\_number = int(input("Minimum value of random numbers:"))

max\_random\_number = int(input("Maximum value of random numbers:"))

list\_random\_number = **create\_list\_random\_number**(no\_of\_random\_number, min\_random\_number, max\_random\_number)

# your code goes here

...

...

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

How many random numbers? 18

Minimum value of random numbers: 11

Maximum value of random numbers: 25

[25, 14, 20, 24, 22, 15, 23, 23, 11, 21, 15, 24, 17, 19, 22, 22, 12, 20]

Input an index you’d like to change: 4

Input the value you’d like to change: 100

[25, 14, 20, 24, **100**, 15, 23, 23, 11, 21, 15, 24, 17, 19, 22, 22, 12, 20]

Input an index you’d like to change:

Hints:

1. To generate random numbers, you should **import random library** and call the **randint (x, y) function**, where the random number generated would be fall within x and y. i.e.

random\_number = random.randint ( x, y )

1. A print function can be called to display all values in the list

After the user input minimum and maximum number of random numbers

After the user input index and value

Answer:

# your code goes here

import random

def create\_list\_random\_number(number, minimum, maximum):

list\_random\_number = list()

for i in range(number):

list\_random\_number.append(random.randint(minimum, maximum))

return list\_random\_number

# Output Generation. You are not allowed to modify the following codes

def main():

no\_of\_random\_number = int(input("How many random numbers?"))

min\_random\_number = int(input("Minimum value of random numbers:"))

max\_random\_number = int(input("Maximum value of random numbers:"))

list\_random\_number = create\_list\_random\_number(no\_of\_random\_number, min\_random\_number, max\_random\_number)

# your code goes here

print(list\_random\_number)

while True:

index = int(input("Input an index you’d like to change:"))

value = int(input("Input the value you’d like to change:"))

list\_random\_number[index] = value

print(list\_random\_number)

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. A list of applicant is stored in a consistent format as shown in the program segment below. Write a program to extract their information display it in a meaningful format as shown below.

# your code goes here

...

...

# Output Generation. You are not allowed to modify the following codes

def main():

applicants = [

"001 kelvinyip@vtc.edu.hk 25952537 Programming Networking Database",

"002 cowleung@nomail.com 98765432 Cloud Security AI BigData",

"003 peterpan@nomail.com 23456789 Web MobileComputing"

]

display\_applicants(applicants)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output:

Total applicants: 3

---

Applicant ID: 001

User Name: kelvinyip

Phone 25952537

This applicant applies for 3 subjects

Subject 1: Programming

Subject 2: Networking

Subject 3: Database

---

Applicant ID: 002

User Name: cowleung

Phone 98765432

This applicant applies for 4 subjects

Subject 1: Cloud

Subject 2: Security

Subject 3: AI

Subject 4: BigData

---

Applicant ID: 003

User Name: peterpan

Phone 23456789

This applicant applies for 2 subjects

Subject 1: Web

Subject 2: MobileComputing

Hints:

1. You may need to use the string split() function more than once
2. Try to make use of the len() function to obtain the total number of applicants or subjects in list instead of using loops to count

The above content is sheer fabrication and any resemblance with any real names herein is purely coincidental. Do not arbitrarily guess and put yourself in someone else’s place.

Answer:

# your code goes here

def display\_applicants(applicants):

print ("Total applicants:", len(applicants))

for i in applicants:

print ("---")

person = i.split(" ")

print ("Applicant ID:",person[0])

print ("User Name:", person[1].split("@")[0])

print ("Phone", person[2])

print ("This applicant applies for "+str(len(person)-3)+" subjects")

for j in range (3, len(person)):

print ("Subject "+str(j-2)+": " + person[j])

# Output Generation. You are not allowed to modify the following codes

def main():

applicants = [

"001 kelvinyip@vtc.edu.hk 25952537 Programming Networking Database",

"002 cowleung@nomail.com 98765432 Cloud Security AI BigData",

"003 peterpan@nomail.com 23456789 Web MobileComputing"

]

display\_applicants(applicants)

if \_\_name\_\_ == "\_\_main\_\_":

main()