Programming Lab

(CSC6L2)

Submitted by:

Abhishek Tandon

Roll Number: 17208

B.Tech. CSE 3rd Year

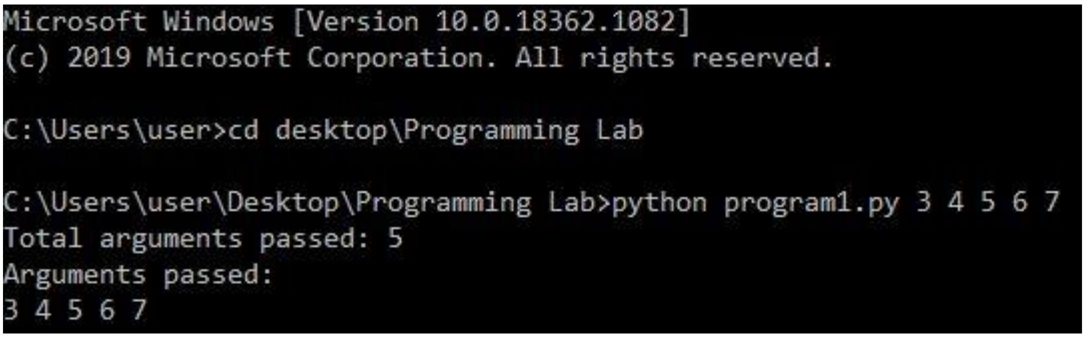
Date: 20/09/2020

Index

1. **Write a python program that takes in command line arguments as input and print the number of arguments**

```
import sys
n = len(sys.argv)
print("Total arguments passed:", n-1)
print("Arguments passed:")
for i in range(1, n):
    print(sys.argv[i], end = " ")
```

Output

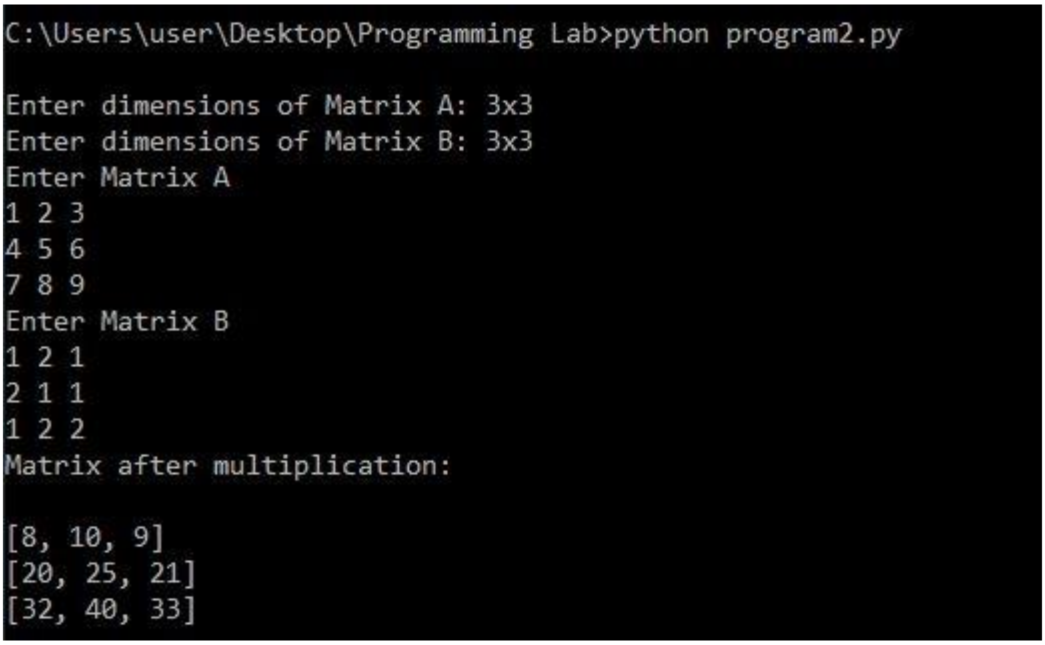


```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd desktop\Programming Lab

C:\Users\user\Desktop\Programming Lab>python program1.py 3 4 5 6 7
Total arguments passed: 5
Arguments passed:
3 4 5 6 7
```

2. Write a python program to perform Matrix Multiplication.

```
M, N = map(int, input("\nEnter dimensions of Matrix A:
").split('x'))
P, Q = map(int, input("Enter dimensions of Matrix B:
").split('x'))
if(N != P):
    print("Matrix Multiplication not possible")
    exit()

print("Enter Matrix A")
A = list()
for i in range(M):
    A.append(list(map(int, input().split(' '))))
print("Enter Matrix B")
B = list()
for i in range(P):
    B.append(list(map(int, input().split(' '))))
result = list()
for i in range(M):
    result.append([0]*Q)

for i in range(len(A)):
    for j in range(len(B[0])):
        for k in range(len(B)):
            result[i][j] += A[i][k] * B[k][j]
print("Matrix after multiplication: \n")

for r in result:
    print(r)
```

Output:

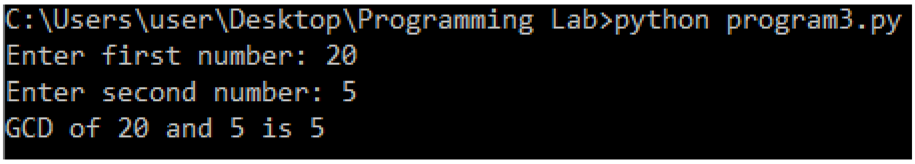


```
C:\Users\user\Desktop\Programming Lab>python program2.py

Enter dimensions of Matrix A: 3x3
Enter dimensions of Matrix B: 3x3
Enter Matrix A
1 2 3
4 5 6
7 8 9
Enter Matrix B
1 2 1
2 1 1
1 2 2
Matrix after multiplication:

[8, 10, 9]
[20, 25, 21]
[32, 40, 33]
```

3. Write a python program to compute GCD of two numbers.

```
def gcd(a,b):
    if (b == 0):
         return a
    return gcd(b, a%b)
a = int(input("Enter first number: ", a))
b = int(input("Enter second number: ", b))
print('GCD of', a, 'and', b, 'is', gcd(a, b))
```

Output:

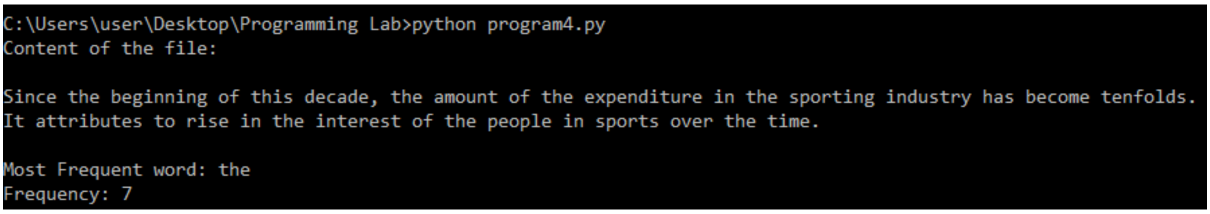


```
C:\Users\user\Desktop\Programming Lab>python program3.py
Enter first number: 20
Enter second number: 5
GCD of 20 and 5 is 5
```

4. Write a python program to find the most frequent words in a text file.

```
f = open('file.txt', 'r')
content = f.read()
print('Content of the file: \n')
print(content)
words = content.split(' ')
dictionary = dict()
for w in words:
    if w not in dictionary.keys():
        dictionary[w.lower()] =  1
    else:
        dictionary[w.lower()] = dictionary[w.lower()] + 1
sortedList = sorted(dictionary.items(), key=lambda x: x[1], reverse = True)
i, j = sortedList[0]
print('\nMost Frequent word:',i,'\nFrequency:',j)
```

Output:

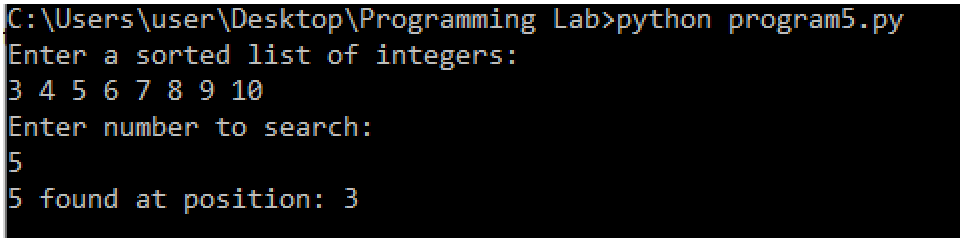


```
C:\Users\user\Desktop\Programming Lab>python program4.py
Content of the file:

Since the beginning of this decade, the amount of the expenditure in the sporting industry has become tenfolds.
It attributes to rise in the interest of the people in sports over the time.

Most Frequent word: the
Frequency: 7
```

5. Write a python program for binary search.

```
def binarySearch(arr, p, r, x):
   if(p <= r):
        q = (p+r)//2
        if arr[q] == x:
            return q
        if arr[q] < x:
            return binarySearch(arr, q+1, r, x)
        if arr[q] > x:
            return binarySearch(arr, p, q-1, x)
    return -1


if __name__ == "__main__":
    arr = list(map(int, input("Enter a sorted list of integers:
\n").split(' ')))
    x = int(input("Enter number to search: \n"))
    val = binarySearch(arr, 0, len(arr)-1, x)
    if val != -1:
        print(x, "found at position:", val+1)
    else:
        print(x, "not found")
```

Output:

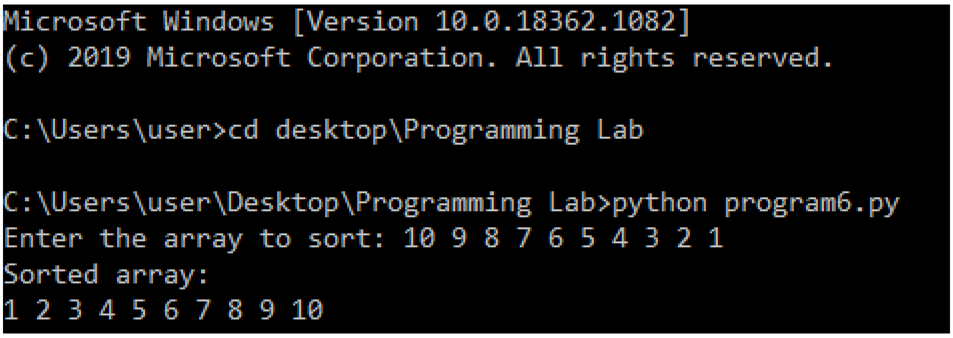


```
C:\Users\user\Desktop\Programming Lab>python program5.py
Enter a sorted list of integers:
3 4 5 6 7 8 9 10
Enter number to search:
5
5 found at position: 3
```

6. Write a python program for Selection Sort

```
arr = list(map(int, input("Enter the array to sort: "). split('
')))


for i in range(len(arr)):
    min_idx = i
    for j in range(i+1, len(arr)):
        if arr[min_idx] > arr[j]:
            min_idx = j
    arr[i], arr[min_idx] = arr[min_idx], arr[i]


print ("Sorted array: ")
for a in arr:
    print(a, end = " ")
```

Output:

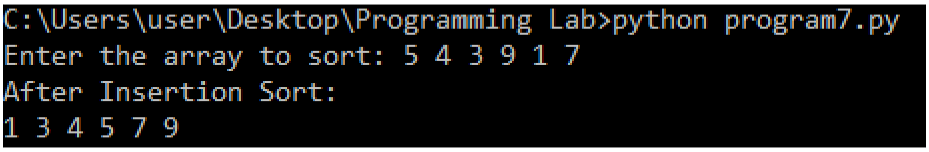


```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd desktop\Programming Lab

C:\Users\user\Desktop\Programming Lab>python program6.py
Enter the array to sort: 10 9 8 7 6 5 4 3 2 1
Sorted array:
1 2 3 4 5 6 7 8 9 10
```

7. Write a python program for Insertion Sort

```
def insertionSort(arr):

    for i in range(len(arr)):

        key = arr[i]

        j = i-1

        while j >= 0 and key < arr[j]:

                arr[j + 1] = arr[j]

                j -= 1

        arr[j + 1] = key


arr = list(map(int, input("Enter the array to sort: "). split(' ')))

insertionSort(arr)

print("After Insertion Sort: ")

for a in arr:

    print (a, end = " ")
```

Output:

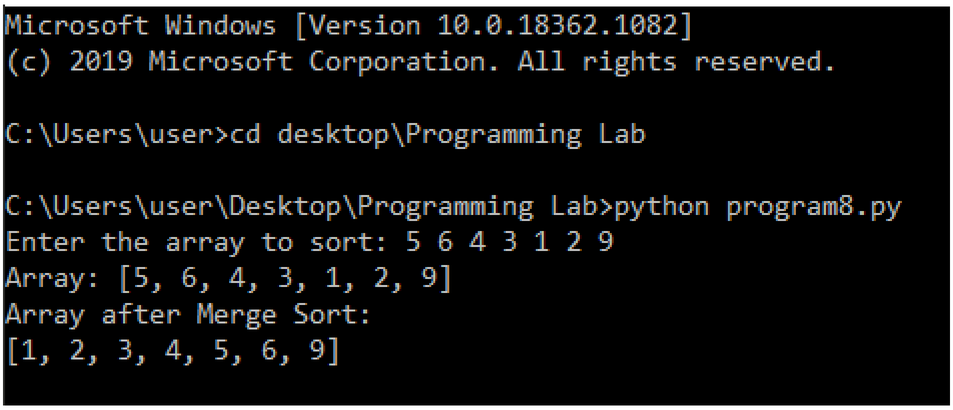


```
C:\Users\user\Desktop\Programming Lab>python program7.py
Enter the array to sort: 5 4 3 9 1 7
After Insertion Sort:
1 3 4 5 7 9
```

8. Write a python program for Merge Sort

```
def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i+= 1
            else:
                arr[k] = R[j]
                j+= 1
            k+= 1
        while i < len(L):
            arr[k] = L[i]
            i+= 1
            k+= 1
        while j < len(R):
            arr[k] = R[j]
            j+= 1
            k+= 1

if __name__ == '__main__':
    arr = list(map(int, input("Enter the array to sort: ").
split(' ')))
    print("Array:",arr)
    mergeSort(arr)
    print("Array after Merge Sort:", end ="\n")
    print(arr)
```

Output:

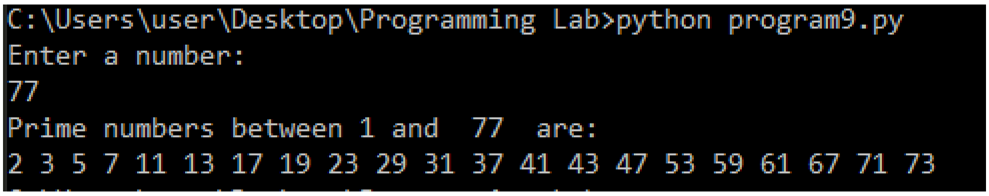


```
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd desktop\Programming Lab

C:\Users\user\Desktop\Programming Lab>python program8.py
Enter the array to sort: 5 6 4 3 1 2 9
Array: [5, 6, 4, 3, 1, 2, 9]
Array after Merge Sort:
[1, 2, 3, 4, 5, 6, 9]
```

9. Write a python program to find first n prime numbers

```
N = int(input("Enter a number: \n"))
i, j, f = 0, 0, 0
print("Prime numbers between 1 and ", N , " are:")
for i in range(1, N + 1, 1):
    if (i == 1 or i == 0):
        continue;
    flag = 1
    for j in range(2, ((i // 2) + 1), 1):
        if (i % j == 0):
            flag = 0;
            break
    if (flag == 1):
        print(i, end = " ")
```

Output:

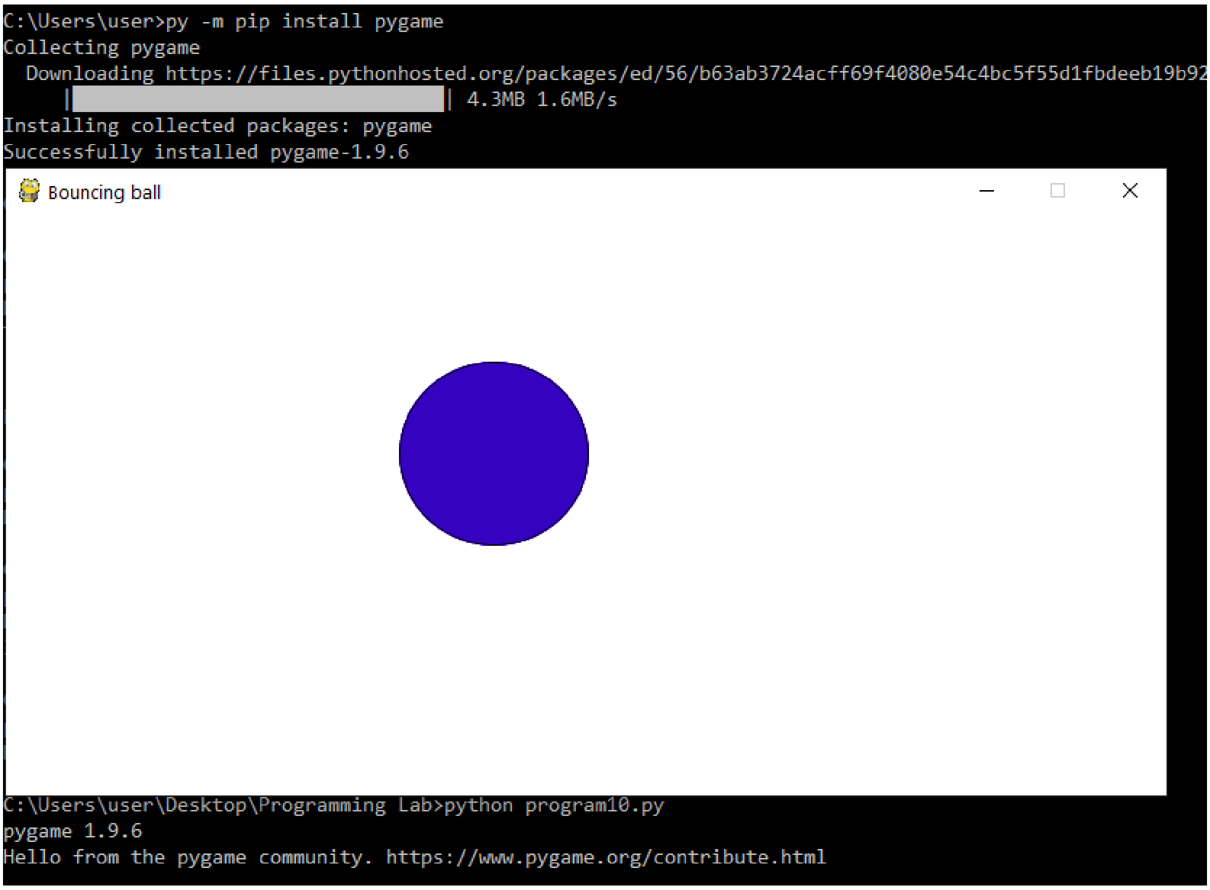


```
C:\Users\user\Desktop\Programming Lab>python program9.py
Enter a number:
77
Prime numbers between 1 and  77  are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73
```

10.Write a python program to simulate bouncing ball in pygame.

```
import sys, pygame
pygame.init()
size = width, height = 800, 400
speed = [1, 1]
background = 255, 255, 255
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Bouncing ball")
ball = pygame.image.load("ball.png")
ballrect = ball.get_rect()
while 1:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]
    screen.fill(background)
    screen.blit(ball, ballrect)
    pygame.display.flip()
```

Output:



```
C:\Users\user>py -m pip install pygame
Collecting pygame
  Downloading https://files.pythonhosted.org/packages/ed/56/b63ab3724acff69f4080e54c4bc5f55d1fbdeeb19b92
     |████████████████████████████████| 4.3MB 1.6MB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.6
```

```
C:\Users\user\Desktop\Programming Lab>python program10.py
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Footnotes:

GitHub repository:

https://github.com/abhishektandon/programming-lab-csc6l2-17208