

**Summer Fields School**

**(ISO 9001:2000, ISO14001:2004)  
DLF Qutab Enclave Complex, Phase-I  
Gurgaon-122001, Haryana, INDIA**

**COMPUTER SCIENCE**

**PRACTICAL FILE**

(SESSION 2019 – 2020)

Submitted by Under the guidance of

**Nishant Tehlan Mrs. Sushma Sharma**

Class: **XII A (Non Medical) Computer Sc. teacher**

Roll No: \_\_\_\_\_\_\_\_\_\_\_

***CERTIFICATE***

This is to certify that **Nishant Tehlan**, student of class**XII A** has successfully completed the Practical Work on**Computer Sc.(Python)** under the guidance of**Mrs. Sushma Sharma**(Computer Sc. Teacher) during the year**2019-20**inpartial fulfilment of the Computer Science Practical examination conducted by **CBSE.**

­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SUBJECT TEACHER’S SIGNATURE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EXTERNAL EXAMINER’S SIGNATURE

***ACKNOWLEDGEMENT***

I am extremely grateful to Mrs. Sushma Sharma, Teacher of Department of Computer Science and our principal Mrs. Irina Mukherjee for their able guidance and useful suggestions, which helped me in completing the practical work in time.

I would also like to thank all the teaching and non-teaching staff of Computer Science Department who helped me directly or indirectly in the completion of this project.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends and classmates for their help and wishes for the successful completion of this project.

**Nishant Tehlan**

Class: **XII A (Non Medical)**

Roll No: \_\_\_\_\_\_\_\_\_\_\_

**Summer Fields School**

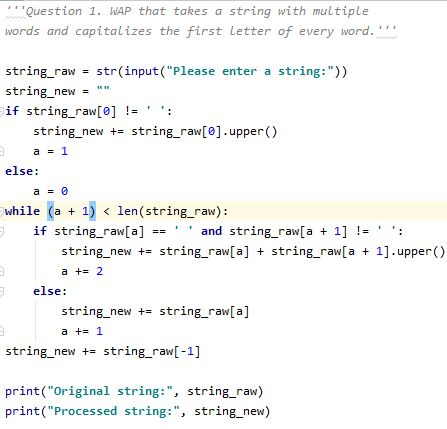
**(ISO 9001:2000, ISO14001:2004)  
DLF Qutab Enclave Complex, Phase-I  
Gurgaon-122001, Haryana, INDIA**

**INDEX**

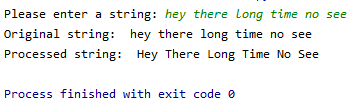
|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Topic** | **Page No.** | **Signature** |
| **1.** | WAP that takes a string with multiple words and capitalizes the first letter of every word. |  |  |
| **2.** | WAP that checks a string if it’s a palindrome or not. |  |  |
| **3.** | WAP that prints the longest word in a list of words. |  |  |
| **4.** | Write a function that receives two parameters **x** and **n** and returns the **n**th root of **x** i.e., **x**1/n. |  |  |
| **5.** | WAP to sort a dictionary’s values using Bubble sort and produce the sorted values as a list. |  |  |
| **6.** | Write a function that takes a number **n** and returns a random number having exactly **n** digits. |  |  |
| **7.** | Write a function that takes two numbers and returns the number that has a lower value in ones place. |  |  |
| **8.** | Write a function that takes first and last values of the series and generates four terms that are equidistant. |  |  |
| **9.** | Create a module MassConversions.py that stores functions for mass conversions e.g.,   * kgtotonne() to convert kg to tonne * tonnetokg() to covert tonne to kg * kgtopound() to convert kg to pound * poundtokg() to convert pound to kg |  |  |
| **10.** | WAP to count the words “to” and “the” present in the text file “Poem.txt”. |  |  |
| **11.** | WAP that appends the contents of one file to another. |  |  |
| **12.** | WAP that reads characters from keyboard and stores all lower case characters in the file LOWER and all the upper case characters in the file UPPER and all the other characters in the file OTHERS. |  |  |
| **13.** | Write a recursive code to compute greatest common divisor of two numbers. |  |  |
| **14.** | Write a function to count and display all the lines starting with alphabet ‘A’ present in a text file “Lines.txt”. |  |  |
| **15.** | Write a recursive code to implement hailstone. |  |  |
| **16.** | WAP that takes a number and checks if it is a happy number by using functions.. |  |  |
| **17.** | Consider the following data given below. Create sequences required from the data below.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Zones | Jan | Feb | Mar | Apr | May | Jun | | North | 140 | 130 | 130 | 190 | 160 | 200 | | South | 160 | 200 | 130 | 200 | 200 | 170 | | East | 140 | 180 | 150 | 170 | 190 | 140 | | West | 180 | 150 | 200 | 120 | 180 | 140 | | Central | 110 | 160 | 130 | 110 | 120 | 170 |   Write code to:   1. Create bar charts to see the distribution of rainfall from Jan to Jun for all zones. 2. Create a pie chart to check to amount of rainfall in Jan separately. 3. Create a line chart to observe any trends from Jan to Jun. |  |  |
| **18.** | Consider the data given below :   |  |  |  | | --- | --- | --- | | **App Name** | **App Price(Rs)** | **Total Downloads** | | Angry Birds | 190 | 197000 | | Teen Titan | 245 | 209000 | | Marvel Comics | 550 | 414000 | | ColorMe | 55 | 196000 | | Fun Run | 175 | 272000 | | Crazy Taxi | 75 | 311000 | | Igram Pro | 140 | 213000 |   Using the above data, plot the following:   1. A line chart depicting the prices of the apps. 2. A bar chart depicting the downloads of the apps. |  |  |
| **19.** | Implement binary search in an array. |  |  |
| **20.** | Implement the linear search in an array. |  |  |
| **21.** | WAP to insert an element in an array with sorted values. |  |  |
| **22.** | WAP to delete an element in an array. |  |  |
| **23.** | A list STL = [‘Meensha Jain’, ‘khushi khan’, ‘Raman Singh’, ‘yashi Sheril’] contains names of students. Names are supposed to be entered with the first letters of their first name and last name capitalized.   1. Selecting only those correctly entered entries. 2. Selecting only the incorrect names. 3. Returning a list with corrected names. |  |  |
| **24.** | WAP to implement stacks in Python. |  |  |
| **25.** | WAP to implement queues in Python. |  |  |
| **26.** | Converting Infix expressions to Postfix expressions using functions push() & pop(). |  |  |
| **27.** | Evaluate Postfix expressions using functions push() & pop(). |  |  |

Practical Questions

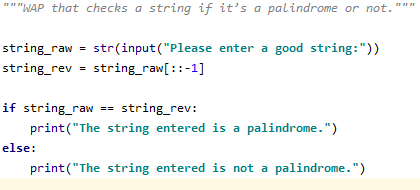
Q1.



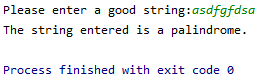
Output:



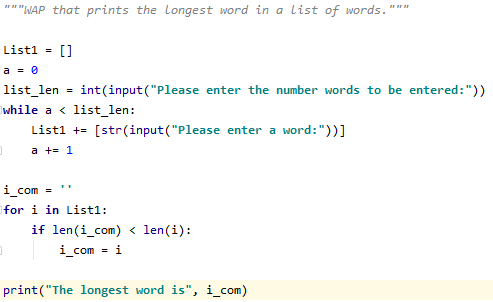
Q2.



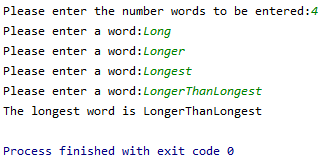
Output:



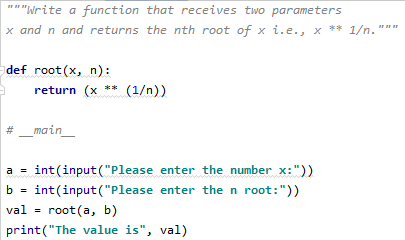
Q3.



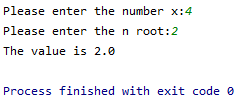
Output:



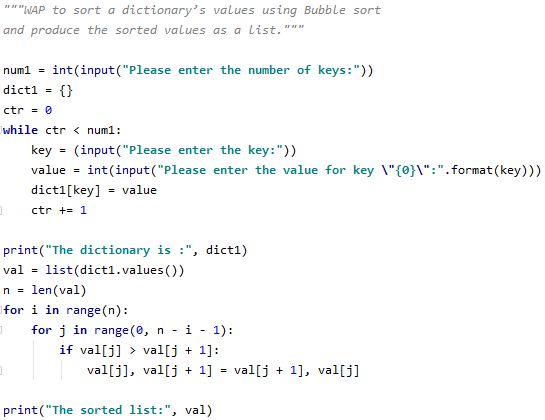
Q4.



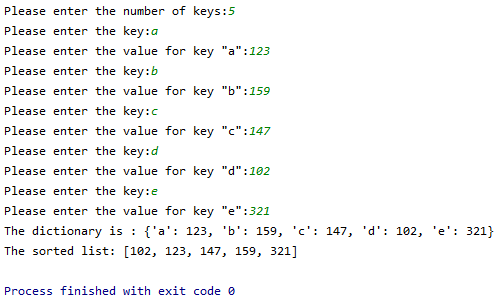
Output:



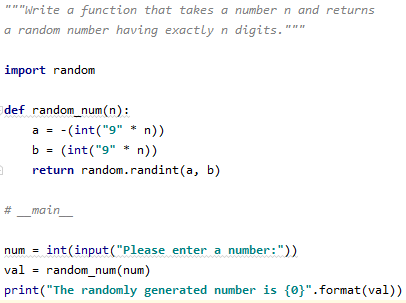
Q5.



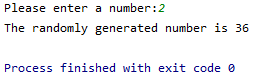
Output:



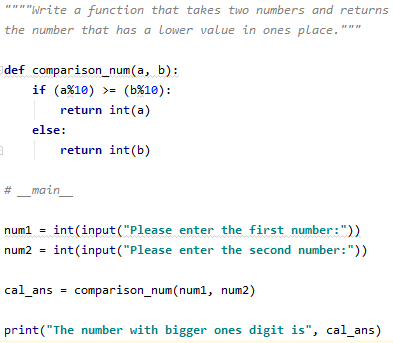
Q6.



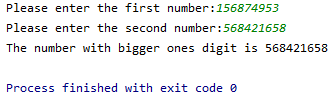
Output:



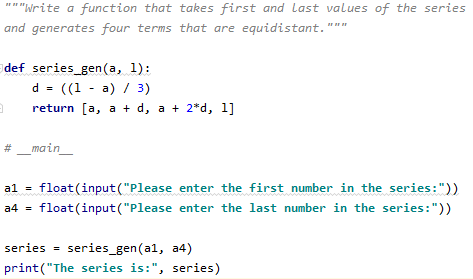
Q7.



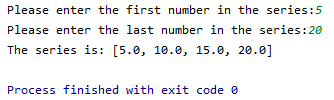
Output:



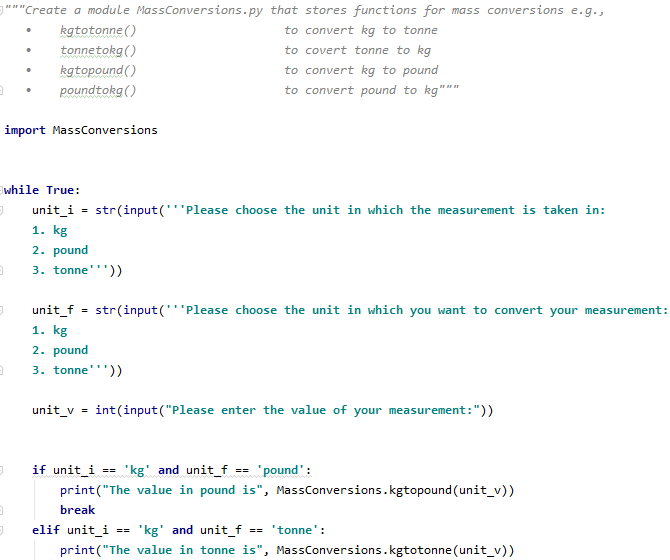
Q8.

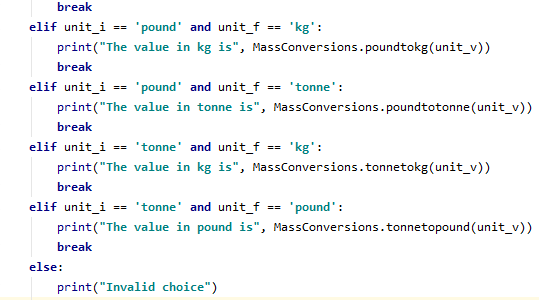


Output:

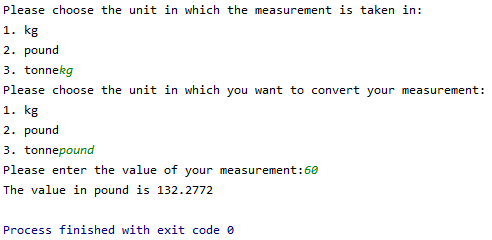


Q9.

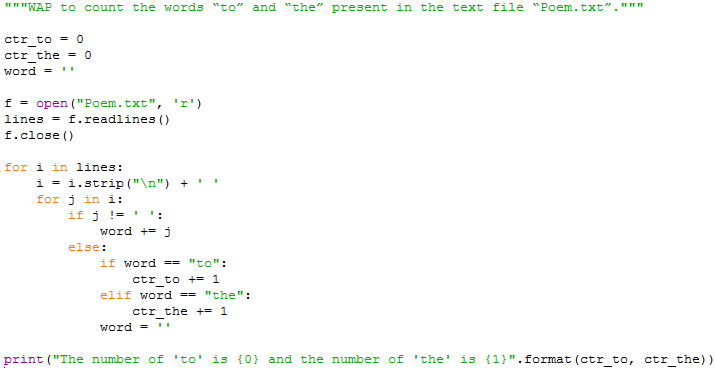




Output:



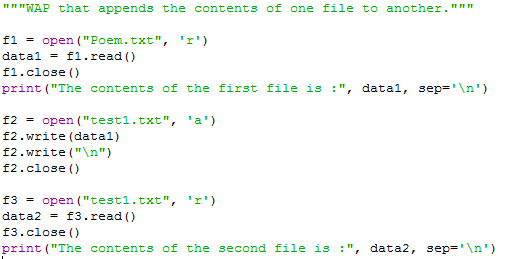
Q10.



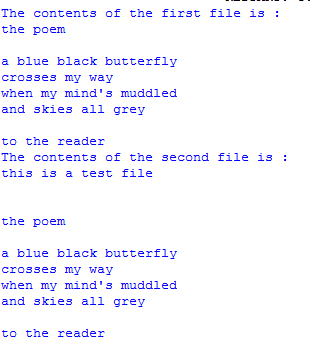
Output:



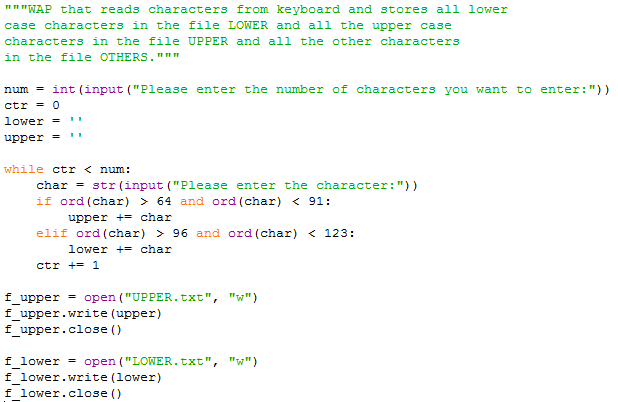
Q11.



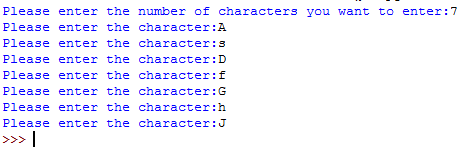
Output:



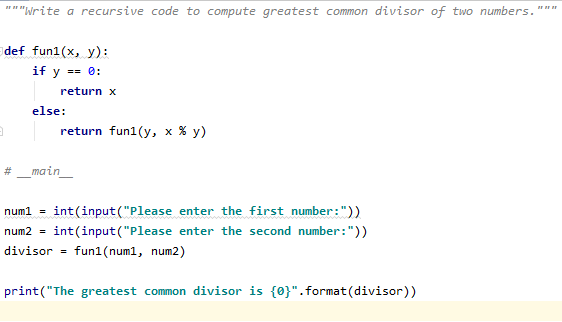
Q12.



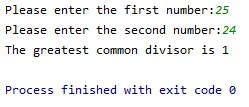
Output:



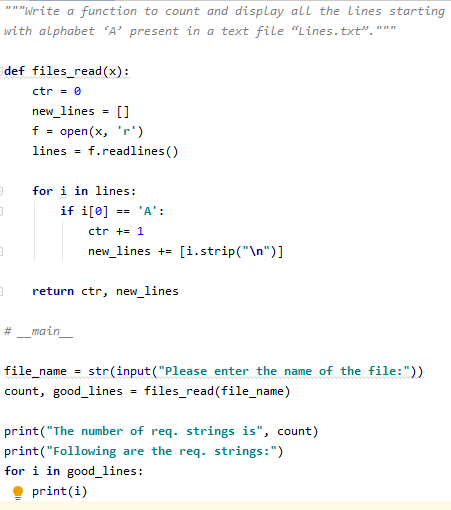
Q13.



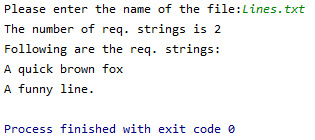
Output:



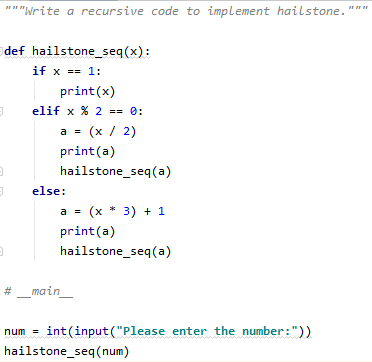
Q14.



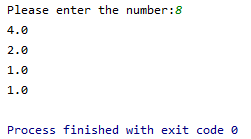
Output:



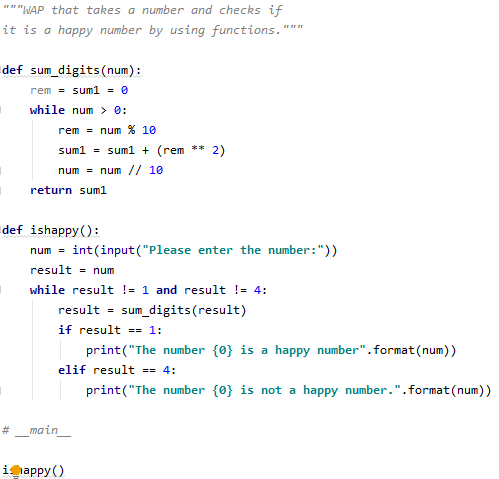
Q15.



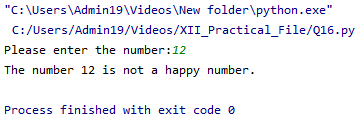
Output:



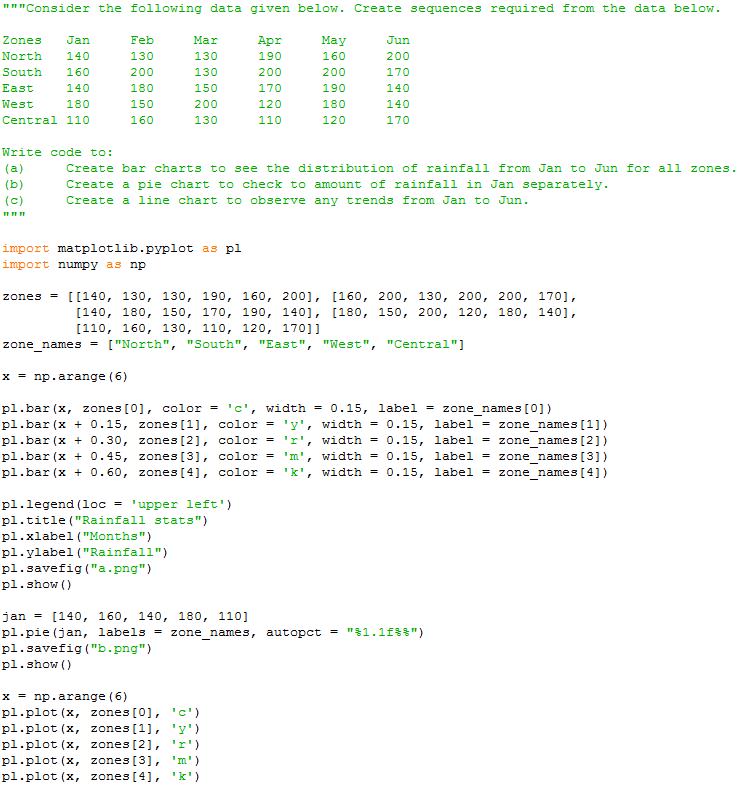
Q16.



Output:

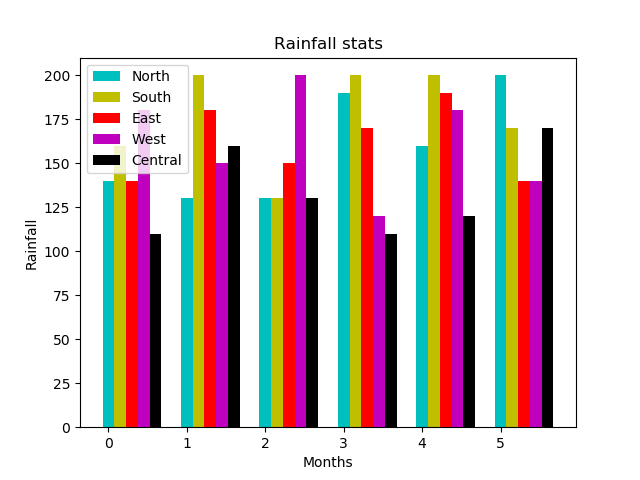


Q17.

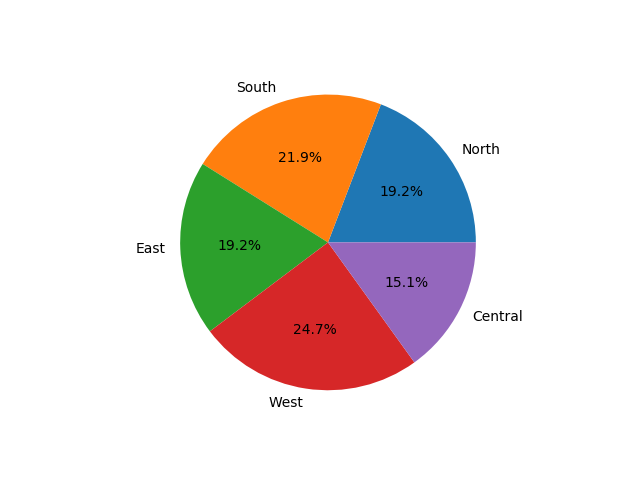


Output:

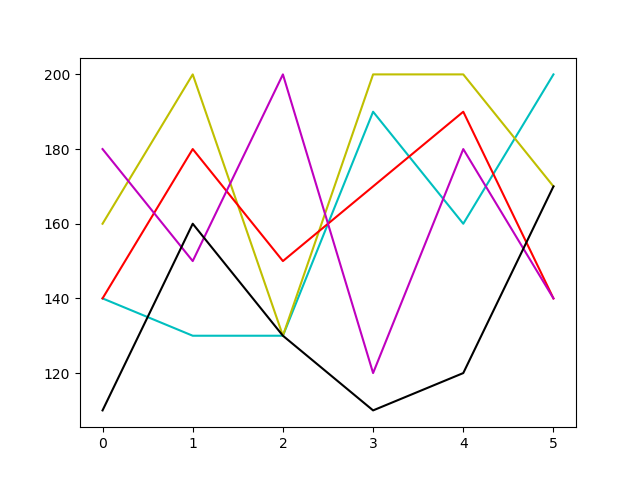
(a)



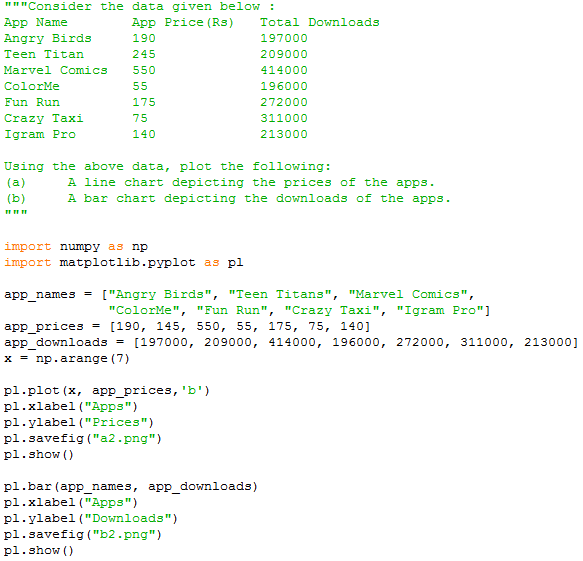
(b)



(c)

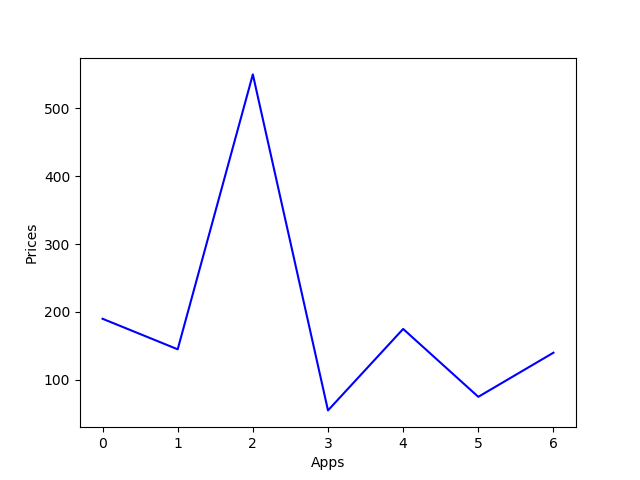


Q18.

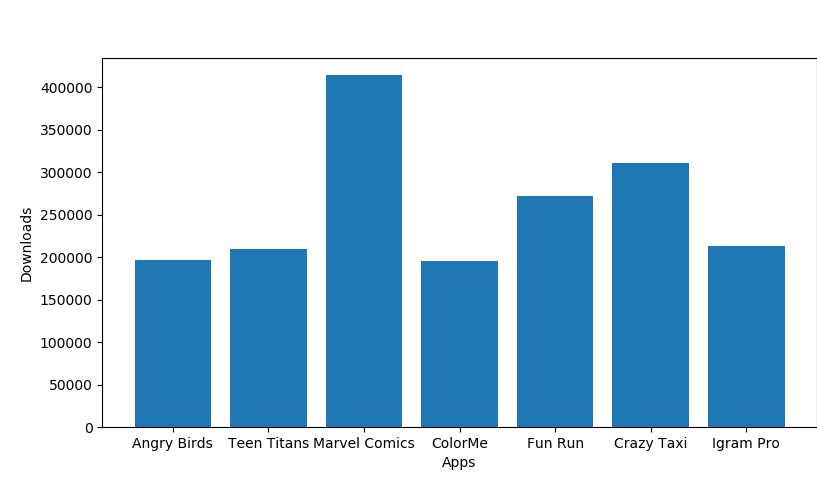


Output:

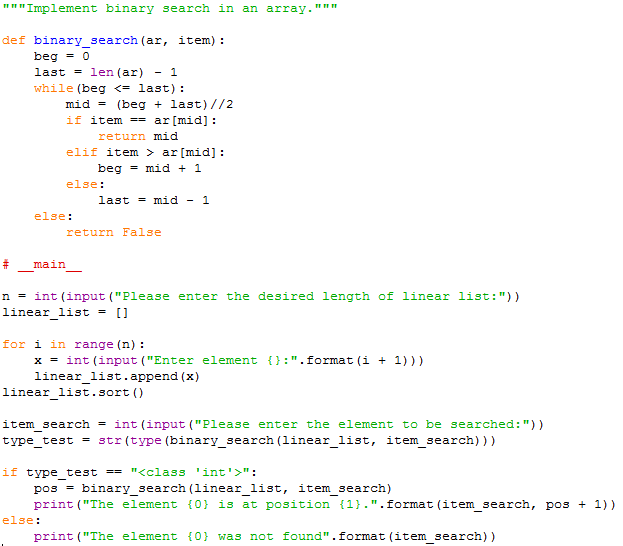
(a)



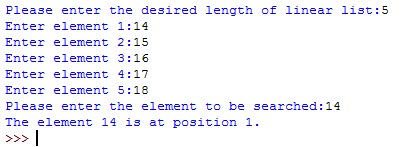
(b)



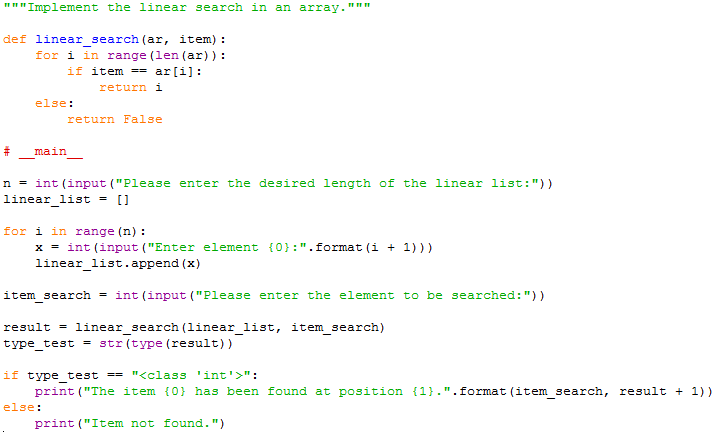
Q19.



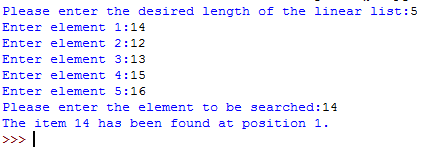
Output:



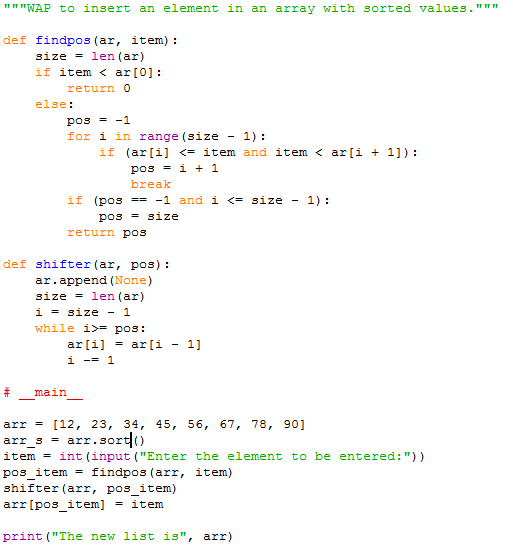
Q20.



Output:



Q21.



Output:



Q22.