Assignment 1 Basic Language Syntax and Simple Charts

Part 1: Basic Python Syntax

1. Create a variable, var, and set the value to the string “Hello, world!”.

In [1]:

```
var='Hello, world!'
var
```

Out[1]:

```
'Hello, world!'
```

2. Now print the length of var.

In [2]:

```
print(len(var))
```

```
13
```

3. Extract “world” from var using string index slicing.

In [3]:

```
x=var.index('w')
y=var.index('d')
var[x:(y+1):1]
```

Out[3]:

```
'world'
```

4.Print the types for 1, 1.0, and “1” and note how they differ.

In [4]:

```
type(1)
```

Out[4]:

```
int
```

In [5]:

```
type(1.0)
```

Out[5]:

```
float
```

In [6]:

```
type("1")
```

Out[6]:

```
str
```

They are of int, float and string datatypes

5. Calculate and print the area of a circle of radius 5 to two decimal places.

In [7]:

```
radius=5
area=(3.14)*(radius)*(radius)
print(round(area,2))
```

```
78.5
```

6. Write a for loop that prints the numbers 0 to 9.

In [8]:

```
for i in range(10):
    print(i,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

7.Write a for loop that prints only even numbers from 10 to 20.

In [9]:

```
for i in range(10,22,2):
    print(i,end=' ')
```

```
10 12 14 16 18 20
```

8. Write an if/else statement that takes a variable var and prints “Greater than 10” if var is greater than 10, “Less than 10” if var is less than 10, “Equal to 10” if var is equal to 10. Make sure to appropriately test for each case.

In [10]:

```
var=eval(input())
if var>10:
    print('Greater than 10')
if var<10:
    print('Less than 10')
if var==10:
    print('Equal to 10')
```

```
-7.9
Less than 10
```

9.Write a function which takes a string as its parameter, and returns the length of the string.

In [11]:

```
def str_len(string):
    x=len(string)
    return x;
str_len(str(input()))
```

```
hello!!
```

Out[11]:

```
7
```

10. Write a function which takes a positive integer n as its parameter, and prints out all odd

numbers from 1 to n.

In [13]:

```
def odd_num(n):
     for i in range(1,n,2):
        print(i,end=' ')

n=int(input())
if(n>0):
    odd_num(n)
else:
    print("please enter positive integer only")
```

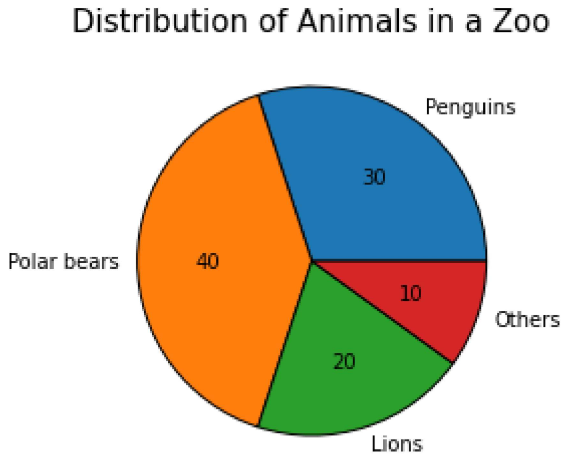
```
9
1 3 5 7
```

Part 2: Simple Charts

1. Pie Chart: Make a pie chart that shows the distribution of animals in a zoo that contains 30 penguins, 40 polar bears, 20 lions, and 10 otters.

In [14]:

```
import matplotlib.pyplot as plt
labels = ['Penguins', 'Polar bears', 'Lions', 'Others']
sizes = [30, 40, 20, 10]
plt.title("Distribution of Animals in a Zoo",fontsize=15)
plt.pie(sizes,labels=labels,wedgeprops={'edgecolor':'black'},autopct='%.0f',r
plt.show()
```

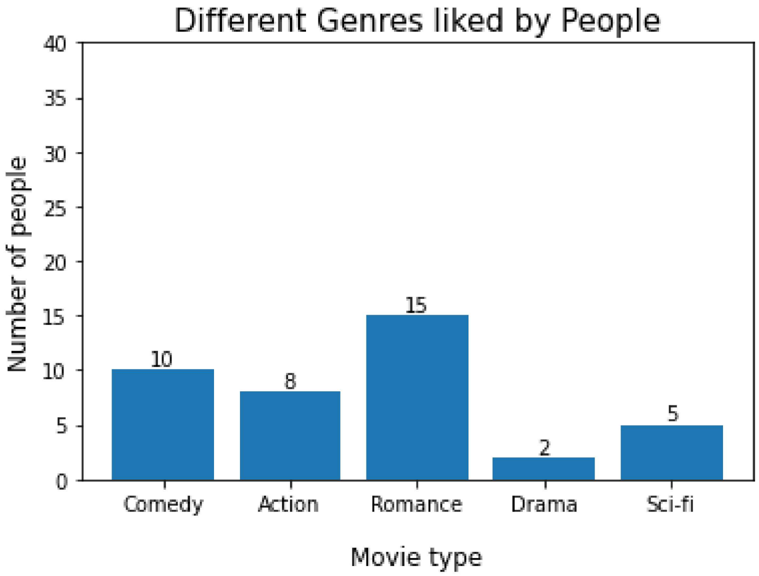


2. Bar Chart: Make a bar chart that shows the number of people, out of a total of 40, who like various movie types if 10 people like comedy; 8 people like action; 15 people like romance; 2 people like drama, and 5 people like sci-fi.

In [15]:

```
import matplotlib.pyplot as plt
import numpy as np

labels = ['Comedy', 'Action', 'Romance', 'Drama','Sci-fi']
sizes = [10, 8,15,2,5]
length=len(labels)
y_axis=np.arange(length)
plt.title("Different Genres liked by People",fontsize=15)
plt.xticks(y_axis,labels,rotation='0')
plt.ylim(0,40)
plt.xlabel('Movie type',fontsize=12,labelpad=15)
plt.ylabel('Number of people',fontsize=12)
genre=plt.bar(y_axis,sizes)
def movie_type(genre):
    for v in genre:
        height = v.get_height()
        plt.text(v.get_x() + v.get_width()/2., height,'%d' % int(height),ha='
movie_type(genre)
plt.show()
```

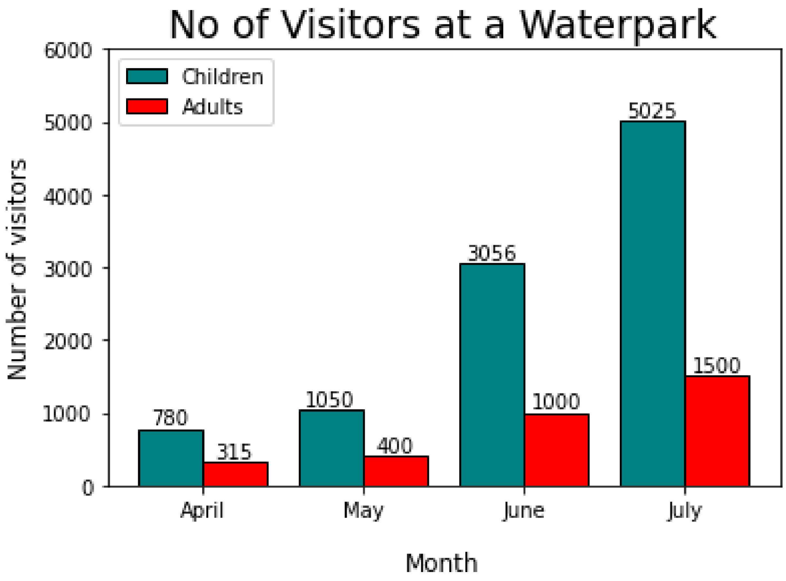


3)Grouped Bar Plots: Make both a side-by-side bar plot and a stacked bar plot that displays the number of child visitors and the number of adult visitors at a waterpark in the months of April, May, June and July. Be sure to include titles, legends and appropriate labels sufficiently sized for readability. April Children: 780 Adults: 315 May Children: 1050 Adults: 400 June Children: 3056 Adults: 1000 July Children: 5025 Adults: 1500

(i) side by side bar plot

In [16]:

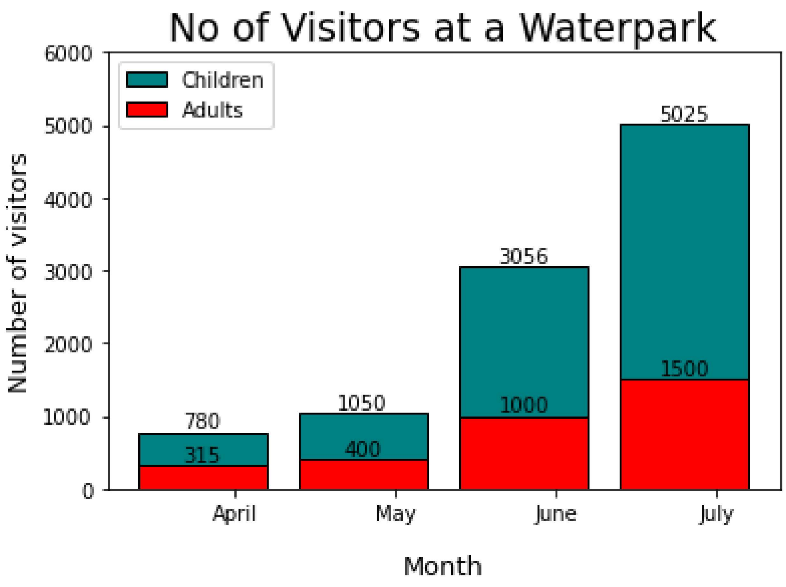
```
import matplotlib.pyplot as plt
import numpy as np
i=np.arange(4)
Month=['April','May','June','July']
Children=[780,1050,3056,5025]
Adults=[315,400,1000,1500]
width=0.4
plt.title("No of Visitors at a Waterpark",fontsize=19)
Children_visitors=plt.bar(i,Children,width,color='teal',label='Children',edge
Adults_visitors=plt.bar(i+width,Adults,width,color='red',label='Adults',edgec
plt.legend(('Children','Adults'))
plt.ylim(0,6000)
plt.xlabel('Month',labelpad=15,fontsize=12)
plt.ylabel('Number of visitors',fontsize=12)
plt.xticks(i+width/2,Month)
def values(visitors):
    for v in visitors:
        height = v.get_height()
        plt.text(v.get_x() + v.get_width()/2., height,'%d' % int(height),ha='
values(Children_visitors)
values(Adults_visitors)
plt.show()
```



(ii) stacked bar plot

In [17]:

```
import matplotlib.pyplot as plt
import numpy as np
i=np.arange(4)
Month=['April','May','June','July']
Children=[780,1050,3056,5025]
Adults=[315,400,1000,1500]
width=0.4
plt.title("No of Visitors at a Waterpark",fontsize=19)
plt.ylim(0,6000)
Children_visitors=plt.bar(Month,Children,label='Children',color='teal',edgeco
Adults_visitors=plt.bar(Month,Adults,label='Adults',color='red',edgecolor='Bl
plt.legend(('Children','Adults'))
plt.xlabel('Month',labelpad=15,fontsize=13)
plt.ylabel('Number of visitors',fontsize=13)
plt.xticks(i+width/2,Month)
def values(visitors):
    for v in visitors:
        height = v.get_height()
        plt.text(v.get_x() + v.get_width()/2., height,'%d' % int(height),ha='
values(Children_visitors)
values(Adults_visitors)
plt.show()
```

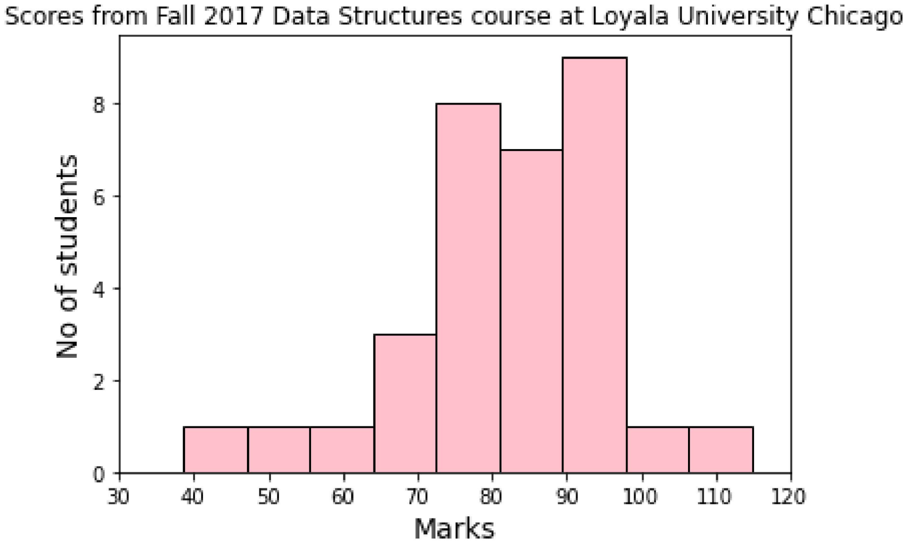


4.Histogram: Make a histogram of the following scores from the Fall 2017 Data Structures course at Loyola University Chicago. Feel free to experiment on the best number of histogram bins for visualization. 114.8, 98.8, 97.3, 96, 94.1, 93.1, 93.1, 91.6, 91.5, 91.3, 90.3, 89.2, 87.5, 87.4, 85.2, 81.7, 81.6, 81.5, 80, 79.3, 78.2, 77.6, 77.1, 76.7, 75.1, 73.9, 72, 71, 64.6, 63.3, 47.2, 38.7

In [18]:

```
import matplotlib.pyplot as plt
import numpy as np
scores=[114.8, 98.8, 97.3, 96, 94.1, 93.1, 93.1, 91.6, 91.5, 91.3, 90.3, 89.2
bins=int(input("enter number of bins you want to have"))
score=plt.hist(scores,bins=bins,edgecolor='black',color='pink')
plt.title('Scores from Fall 2017 Data Structures course at Loyala University
plt.ylabel('No of students',fontsize=14)
plt.xlim(30,120)
plt.xlabel('Marks',fontsize=14)
plt.show()
```

```
enter number of bins you want to have9
```



In [ ]: