In [7]:

```
# Q7 system of linear eqns
import numpy as np
import scipy.linalg as la
A = np.array([[5,2,1],[2,6,-1],[1,-1,3]])
B = np.array([1,0,0])
I = la.solve(A,B)
print("V = ", 2*(I[0]+I[1]))
```

```
V =  0.31746031746031755
```

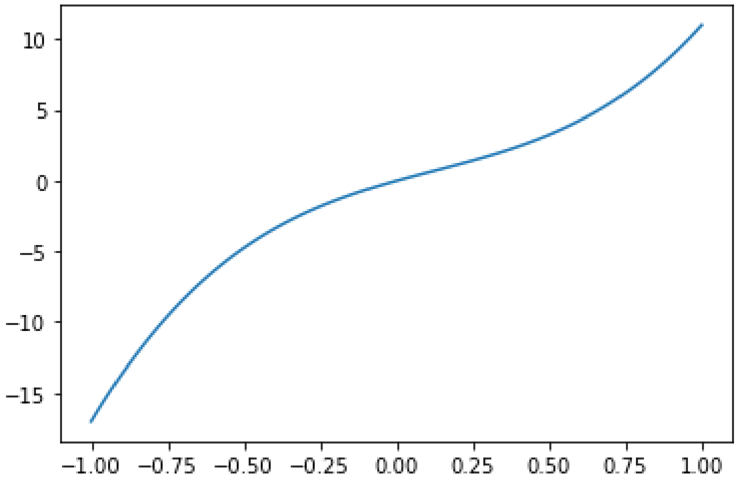
In [12]:

```
# Q11 Roots of a polynomial, to be cross checked
import numpy as np
roots = np.roots ([1,0,1,0,1,1])
for i in roots :
    if i <=1 and i >=-1 :
        print(i)
```

```
(0.7077287898043153+0.8419548540323738j)
(0.7077287898043153-0.8419548540323738j)
(-0.3892873314033924+1.070675774890508j)
(-0.3892873314033924-1.070675774890508j)
(-0.6368829168018451+0j)
```

In [50]:

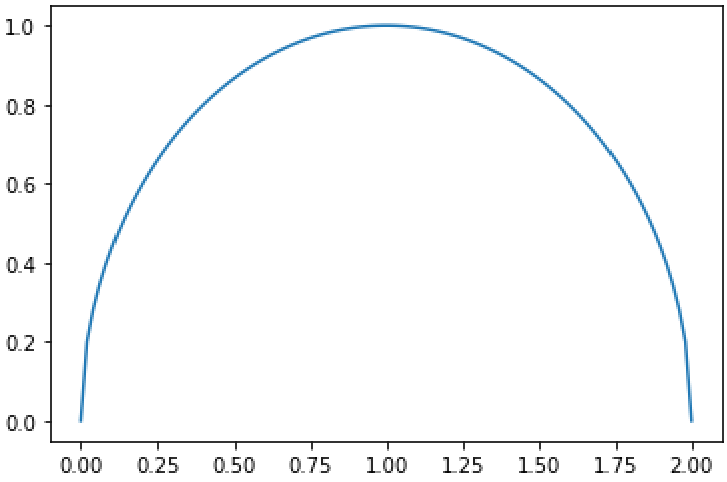
```
# Q12 Plot derivative of polynomial
import numpy.polynomial as poly
import matplotlib.pyplot as plt
p = poly.Polynomial([1,0,3,-1,2])
p_deriv = p.deriv()
t = np.linspace(-1,1,100)
plt.plot(t,p_deriv(t))
plt.show()

# Find  the local minima of the polynomial
ex = p_deriv.roots()
p_ex = np.array([])
for i in ex:
    # To only take roots in range [-1,1]
    if -1<= i and i<=1:
        p_ex = np.append(p_ex,i)
# Find the second derivative
p_deriv2 = p_deriv.deriv()
for i in p_ex :
    # For local minima, p'' (x)>0
    if p_deriv2(i)>0:
        print("local minima at t = ",i)
        print("Value of p(t) at local minima is ", p(i))
```



```
local minima at t =  0j
Value of p(t) at local minima is  (1+0j)
```

In [64]:

```python
# Q14 plot a function, approx the value of pi
import matplotlib.pyplot as plt
import numpy as np
import scipy.integrate as intg
def f(x):
    return np.sqrt(1-((x-1)**2))
t = np.linspace(0,2,100)
y = f(t)
plt.plot(t,y)
plt.show()
# Calculate the area under the curve
area = intg.trapz(y,t)
print("Area under the curve is : ",area)
r = 1
pi = 2 * area / r **2
print("Approx value of pi is : ",pi)
```



```
Area under the curve is :  1.5691090196009048
Approx value of pi is :  3.1382180392018095
```

In [81]:

```python
# Code to generate CSV files.
f = open("file1.csv","w")
f2 = open("file2.csv","w")

age = np.random.randint(10,30,50)
age2 = np.random.randint(8,25,30)
a,b = np.unique(age,return_counts = True)
c,d = np.unique(age2,return_counts = True)
print(a,b)
```

```
for i in np.arange(len(a)):
    f.write ("{},{}\n".format(a[i],b[i]))

for i in np.arange(len(c)):
    f2.write ("{},{}\n".format(c[i],d[i]))
f.close()
f2.close()
```

```
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 29] [5 2 5 1 3 3 1 2 1
2 3 1 5 2 2 3 5 1 3]
```

In [97]:

```
# Q16
import numpy as np
import matplotlib.pyplot as plt
f1 = np.genfromtxt("file1.csv",delimiter=',')
f2 = np.genfromtxt("file2.csv",delimiter=',')
ages1 = f1[:,0]
ages2 = f2[:,0]
ages = np.append(ages1,ages2)
print(ages2)
for i in ages:
    if i in ages1 and i in ages2:
```

```
[ 8.  9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 20. 22. 23.]
10.0 is present
11.0 is present
12.0 is present
13.0 is present
14.0 is present
15.0 is present
16.0 is present
17.0 is present
18.0 is present
19.0 Not present
20.0 is present
21.0 Not present
22.0 is present
23.0 is present
24.0 Not present
25.0 Not present
26.0 Not present
27.0 Not present
29.0 Not present
```

In [91]:

```
a = np.array([1,2,3,4])
6 in a
```

Out[91]:

```
False
```

In [121…

```
a = np.random.randint(6,size=(5,5))
# a[start_row:end_row,startcol:end_col]
print(a)
a[4,4]
```

```
[[2 5 0 1 3]
 [1 0 3 0 5]
 [0 1 0 5 4]
 [1 2 1 1 3]
 [2 5 5 5 0]]
```

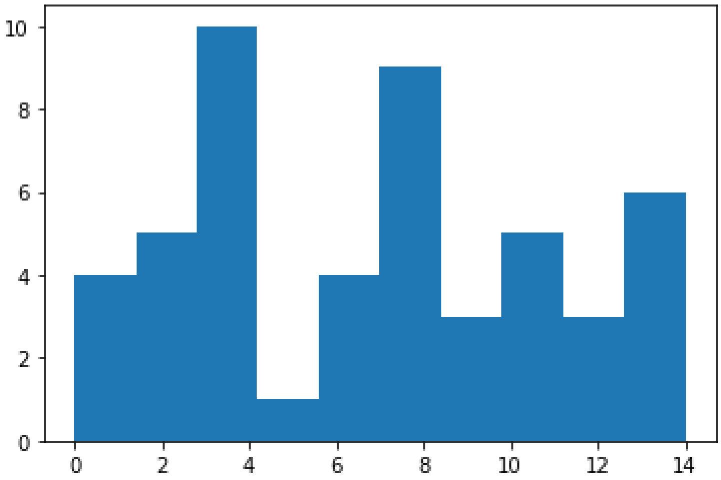
Out[121…

```
0
```

```
In [122…  a = np.random.randint(15,size = 50)
```

```
In [128…  plt.hist(a)
```

```
Out[128…  (array([ 4.,  5., 10.,  1.,  4.,  9.,  3.,  5.,  3.,  6.]),
           array([ 0. ,  1.4,  2.8,  4.2,  5.6,  7. ,  8.4,  9.8, 11.2, 12.6, 14. ]),
           <BarContainer object of 10 artists>)
```



```
In [ ]:
```