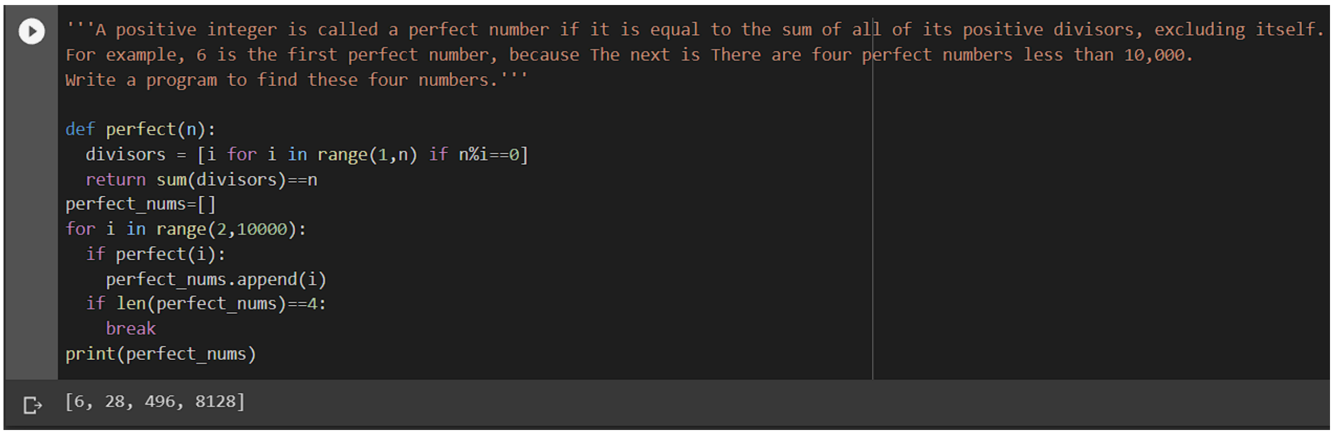
APP WEEK-1 LAB

Q1.
A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself.
For example, 6 is the first perfect number, because The next is There are four perfect numbers less than 10,000.
Write a program to find these four numbers.

Code:

```
def perfect(n):
  divisors = [i for i in range(1,n) if n%i==0]
  return sum(divisors)==n
perfect_nums=[]
for i in range(2,10000):
  if perfect(i):
    perfect_nums.append(i)
  if len(perfect_nums)==4:
    break
print(perfect_nums)
```

SnapShot:



```
'''A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself.
For example, 6 is the first perfect number, because The next is There are four perfect numbers less than 10,000.
Write a program to find these four numbers.'''

def perfect(n):
  divisors = [i for i in range(1,n) if n%i==0]
  return sum(divisors)==n
perfect_nums=[]
for i in range(2,10000):
  if perfect(i):
    perfect_nums.append(i)
  if len(perfect_nums)==4:
    break
print(perfect_nums)
```

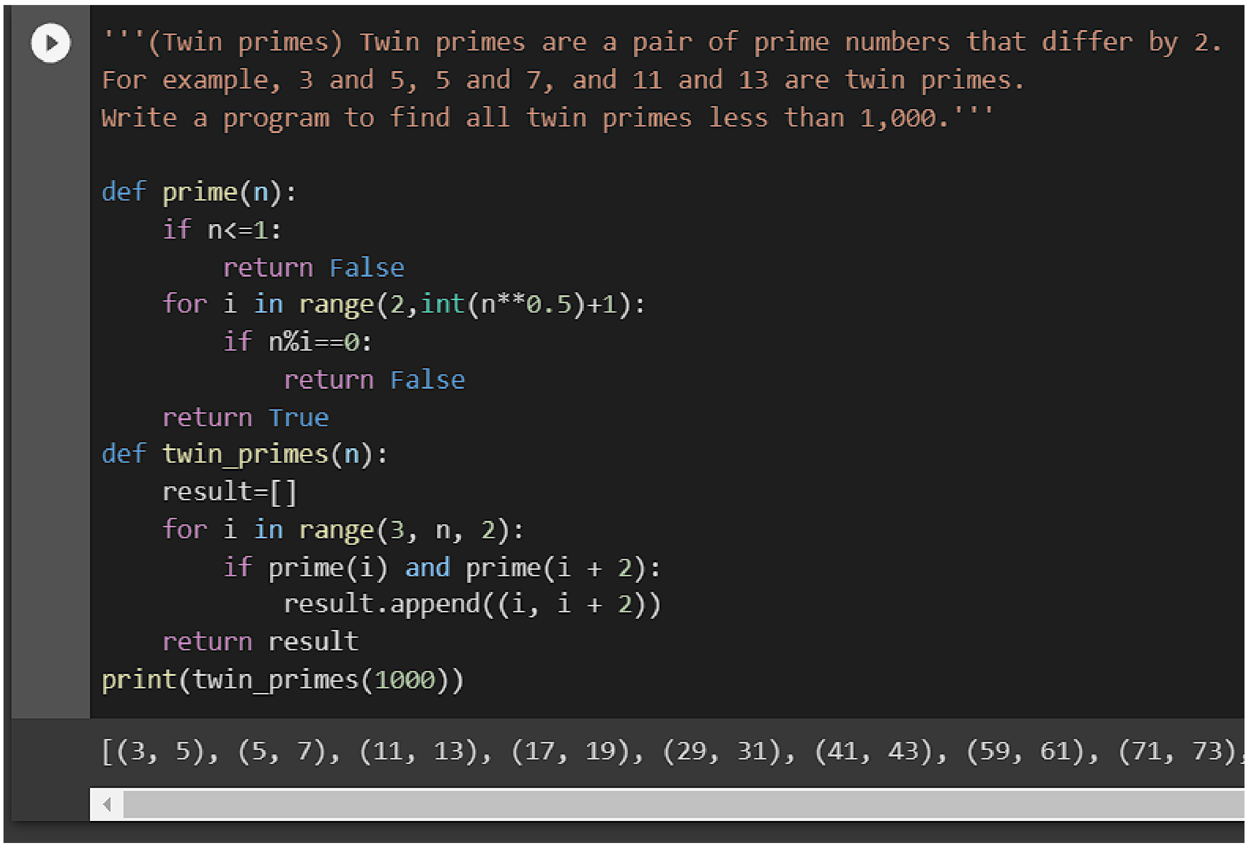
```
[6, 28, 496, 8128]
```

Q2.
(Twin primes) Twin primes are a pair of prime numbers that differ by 2.
For example, 3 and 5, 5 and 7, and 11 and 13 are twin primes.
Write a program to find all twin primes less than 1,000.

<u>Code:</u>

```
def prime(n):
    if n<=1:
        return False
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False
    return True
def twin_primes(n):
    result=[]
    for i in range(3, n, 2):
        if prime(i) and prime(i + 2):
            result.append((i, i + 2))
    return result
print(twin_primes(1000))
```

<u>SnapShot:</u>



```
'''(Twin primes) Twin primes are a pair of prime numbers that differ by 2.
For example, 3 and 5, 5 and 7, and 11 and 13 are twin primes.
Write a program to find all twin primes less than 1,000.'''

def prime(n):
    if n<=1:
        return False
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False
    return True
def twin_primes(n):
    result=[]
    for i in range(3, n, 2):
        if prime(i) and prime(i + 2):
            result.append((i, i + 2))
    return result
print(twin_primes(1000))
```

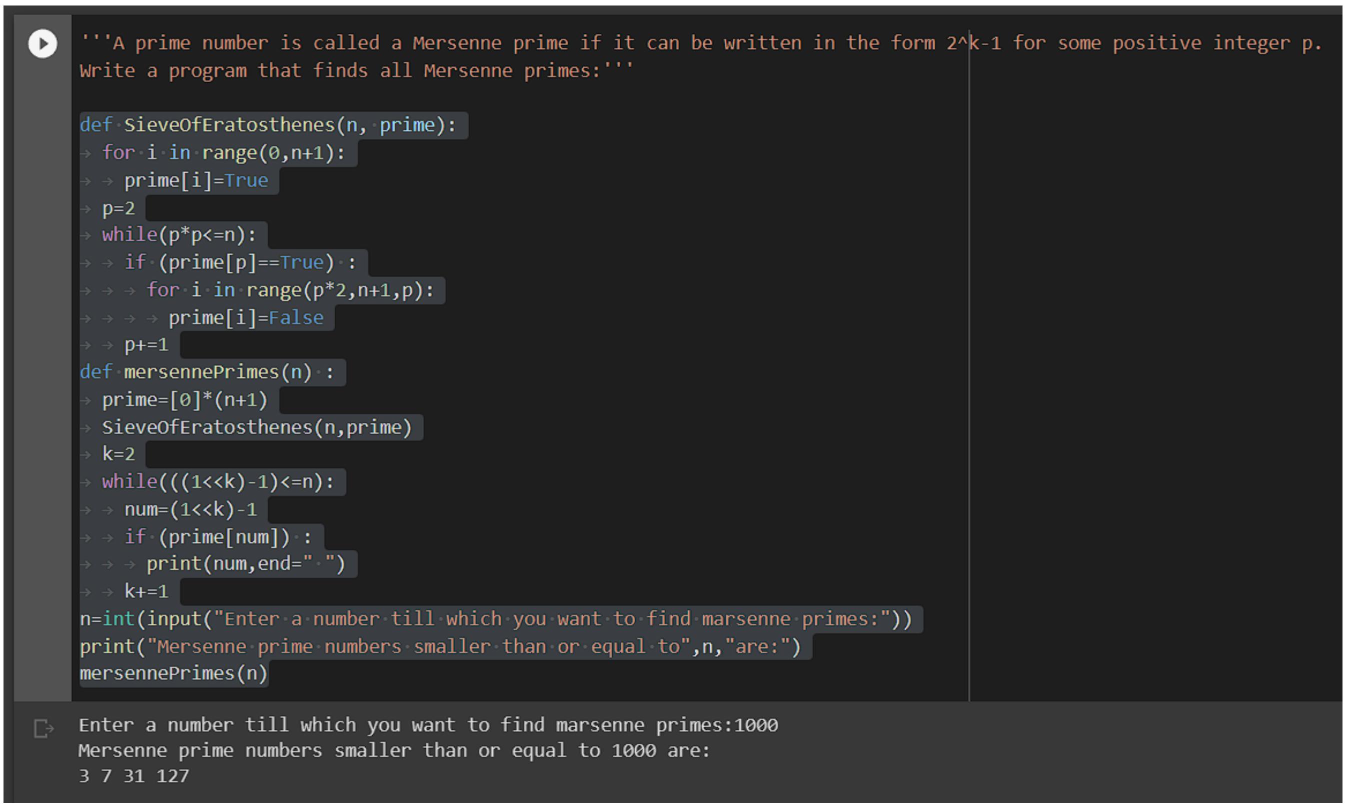
```
[(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73),
```

Q3.
A prime number is called a Mersenne prime if it can be written in the form 2^k-1 for some positive integer p.
Write a program that finds all Mersenne primes

Code:

```
def SieveOfEratosthenes(n, prime):
  for i in range(0,n+1):
    prime[i]=True
  p=2
  while(p*p<=n):
    if (prime[p]==True) :
      for i in range(p*2,n+1,p):
        prime[i]=False
    p+=1
def mersennePrimes(n) :
  prime=[0]*(n+1)
  SieveOfEratosthenes(n,prime)
  k=2
  while(((1<<k)-1)<=n):
    num=(1<<k)-1
    if (prime[num]) :
      print(num,end=" ")
    k+=1
n=int(input("Enter a number till which you want to find marsenne primes:"))
print("Mersenne prime numbers smaller than or equal to",n,"are:")
mersennePrimes(n)
```

SnapShot:



```
'''A prime number is called a Mersenne prime if it can be written in the form 2^k-1 for some positive integer p.
Write a program that finds all Mersenne primes:'''

def SieveOfEratosthenes(n, prime):
	for i in range(0,n+1):
		prime[i]=True
	p=2
	while(p*p<=n):
		if (prime[p]==True) :
			for i in range(p*2,n+1,p):
				prime[i]=False
		p+=1
def mersennePrimes(n) :
	prime=[0]*(n+1)
	SieveOfEratosthenes(n,prime)
	k=2
	while(((1<<k)-1)<=n):
		num=(1<<k)-1
		if (prime[num]) :
			print(num,end=" ")
		k+=1
n=int(input("Enter a number till which you want to find marsenne primes:"))
print("Mersenne prime numbers smaller than or equal to",n,"are:")
mersennePrimes(n)
```

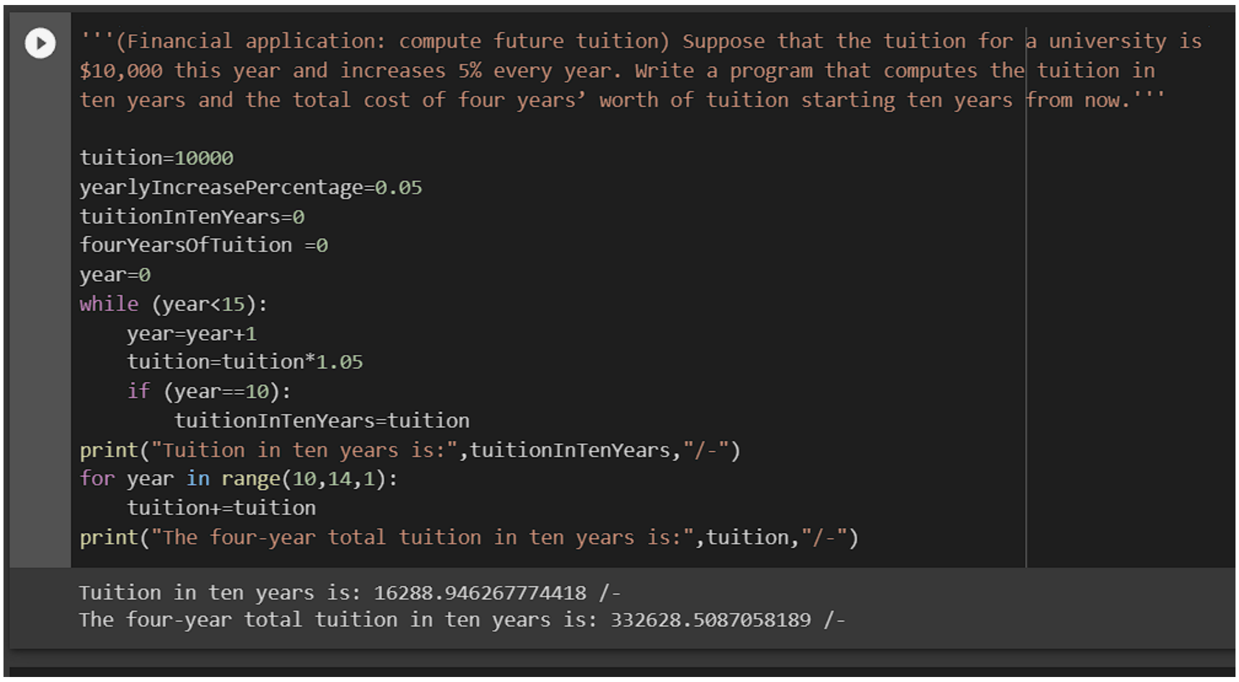
```
Enter a number till which you want to find marsenne primes:1000
Mersenne prime numbers smaller than or equal to 1000 are:
3 7 31 127
```

Q4.
(Financial application: compute future tuition) Suppose that the tuition for a university is $10,000 this year and increases 5% every year. Write a program that computes the tuition in ten years and the total cost of four years' worth of tuition starting ten years from now.

<u>Code:</u>

```
tuition=10000
yearlyIncreasePercentage=0.05
tuitionInTenYears=0
fourYearsOfTuition =0
year=0
while (year<15):
    year=year+1
    tuition=tuition*1.05
    if (year==10):
        tuitionInTenYears=tuition
print("Tuition in ten years is:",tuitionInTenYears,"/-")
for year in range(10,14,1):
    tuition+=tuition
print("The four-year total tuition in ten years is:",tuition,"/-")
```

<u>SnapShot:</u>



```
'''(Financial application: compute future tuition) Suppose that the tuition for a university is
$10,000 this year and increases 5% every year. Write a program that computes the tuition in
ten years and the total cost of four years' worth of tuition starting ten years from now.'''

tuition=10000
yearlyIncreasePercentage=0.05
tuitionInTenYears=0
fourYearsOfTuition =0
year=0
while (year<15):
    year=year+1
    tuition=tuition*1.05
    if (year==10):
        tuitionInTenYears=tuition
print("Tuition in ten years is:",tuitionInTenYears,"/-")
for year in range(10,14,1):
    tuition+=tuition
print("The four-year total tuition in ten years is:",tuition,"/-")
```

```
Tuition in ten years is: 16288.946267774418 /-
The four-year total tuition in ten years is: 332628.5087058189 /-
```

Q5.

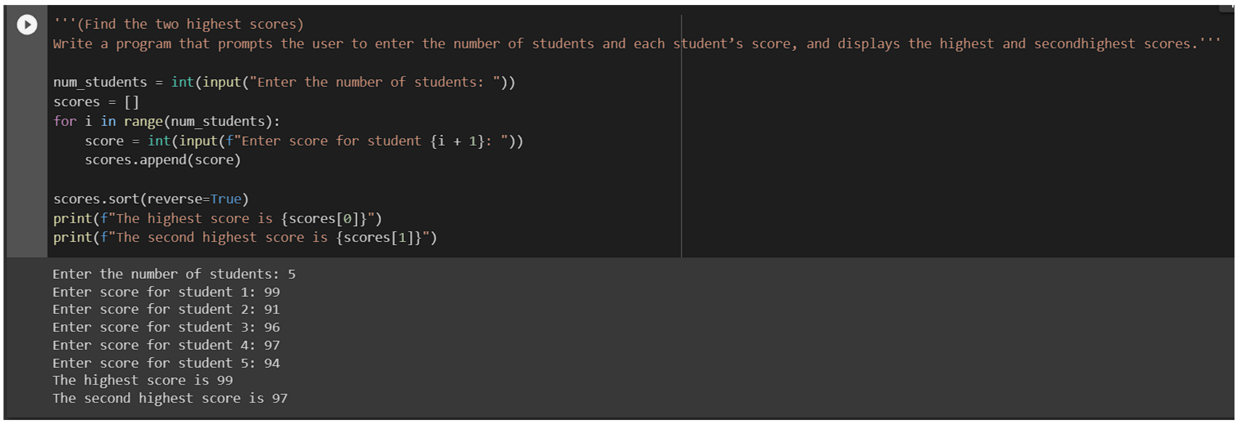
(Find the two highest scores)
Write a program that prompts the user to enter the number of students andeach student's score, and displays the highest and secondhighest scores.

Code:

```
num_students = int(input("Enter the number of students: "))
scores = []
for i in range(num_students):
    score = int(input(f"Enter score for student {i + 1}: "))
    scores.append(score)

scores.sort(reverse=True)
print(f"The highest score is {scores[0]}")
print(f"The second highest score is {scores[1]}")
```

SnapShot:



```
'''(Find the two highest scores)
Write a program that prompts the user to enter the number of students and each student's score, and displays the highest and secondhighest scores.'''

num_students = int(input("Enter the number of students: "))
scores = []
for i in range(num_students):
    score = int(input(f"Enter score for student {i + 1}: "))
    scores.append(score)

scores.sort(reverse=True)
print(f"The highest score is {scores[0]}")
print(f"The second highest score is {scores[1]}")
```

```
Enter the number of students: 5
Enter score for student 1: 99
Enter score for student 2: 91
Enter score for student 3: 96
Enter score for student 4: 97
Enter score for student 5: 94
The highest score is 99
The second highest score is 97
```