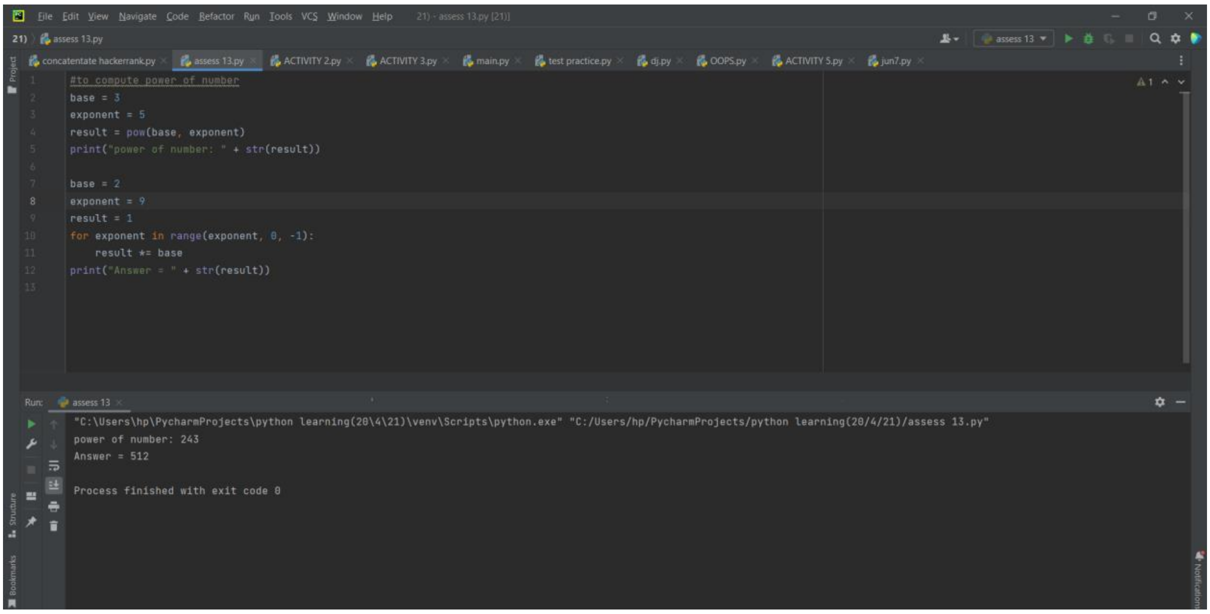
ANSWERS:

```
#1.to compute power of number
base = 3
exponent = 5
result = pow(base, exponent)
print("power of number: " + str(result))
```

another method:

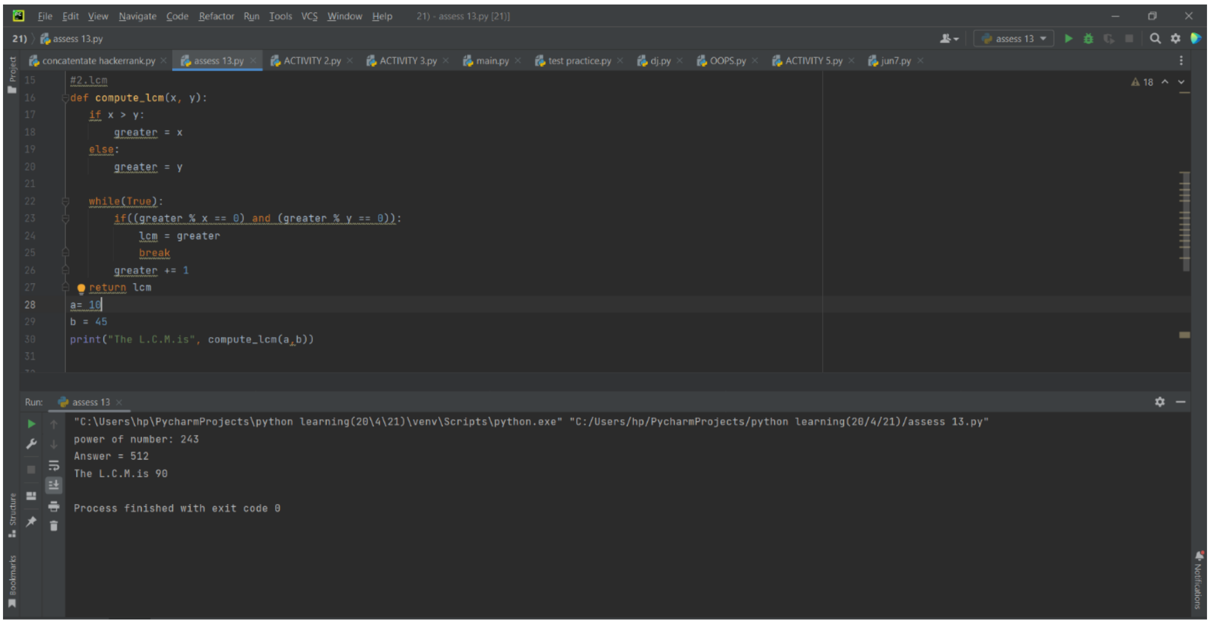
```
base = 2
exponent = 9
result = 1
for exponent in range(exponent, 0, -1):
    result *= base
print("Answer = " + str(result))
```



```
#to compute power of number
base = 3
exponent = 5
result = pow(base, exponent)
print("power of number: " + str(result))

base = 2
exponent = 9
result = 1
for exponent in range(exponent, 0, -1):
    result *= base
print("Answer = " + str(result))
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/assess 13.py"
power of number: 243
Answer = 512

Process finished with exit code 0
```

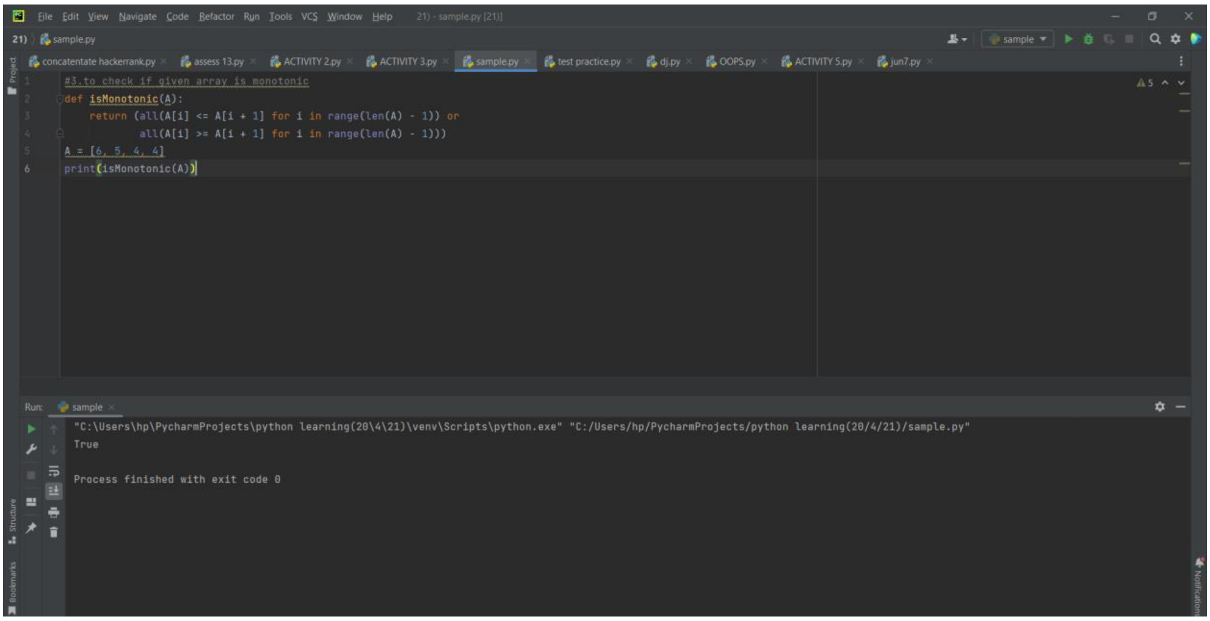
```
#2.lcm
def compute_lcm(x, y):
   if x > y:
       greater = x
   else:
       greater = y

   while(True):
       if((greater % x == 0) and (greater % y == 0)):
           lcm = greater
           break
       greater += 1
   return lcm
a= 10
b = 45
print("The L.C.M.is", compute_lcm(a,b))
```



```
#2.lcm
def compute_lcm(x, y):
    if x > y:
        greater = x
    else:
        greater = y

    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1
    return lcm
a= 10
b = 45
print("The L.C.M.is", compute_lcm(a,b))
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/assess 13.py"
power of number: 243
Answer = 512
The L.C.M.is 90

Process finished with exit code 0
```

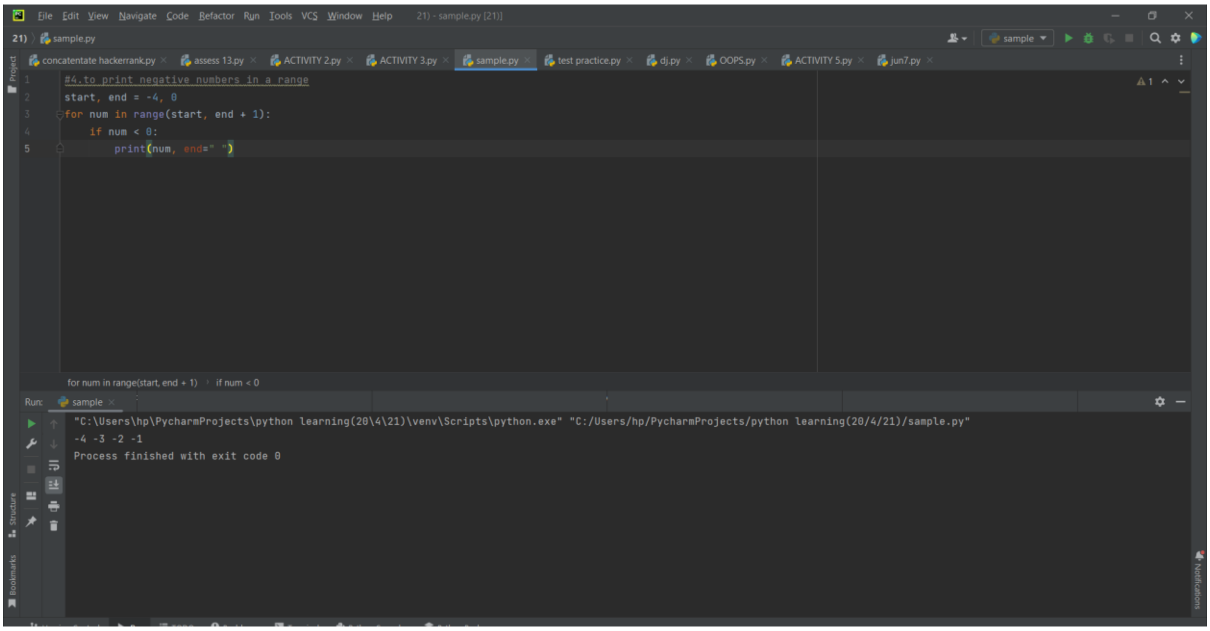
```
#3.to check if given array is monotonic
def isMonotonic(A):
    return (all(A[i] <= A[i + 1] for i in range(len(A) - 1)) or
            all(A[i] >= A[i + 1] for i in range(len(A) - 1)))
A = [6, 5, 4, 4]
print(isMonotonic(A))
```



```
#3.to check if given array is monotonic
def isMonotonic(A):
    return (all(A[i] <= A[i + 1] for i in range(len(A) - 1)) or
            all(A[i] >= A[i + 1] for i in range(len(A) - 1)))
A = [6, 5, 4, 4]
print(isMonotonic(A))
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/sample.py"
True

Process finished with exit code 0
```

```
#4.to print negative numbers in a range
start, end = -4, 0
for num in range(start, end + 1):
    if num < 0:
        print(num, end=" ")
```

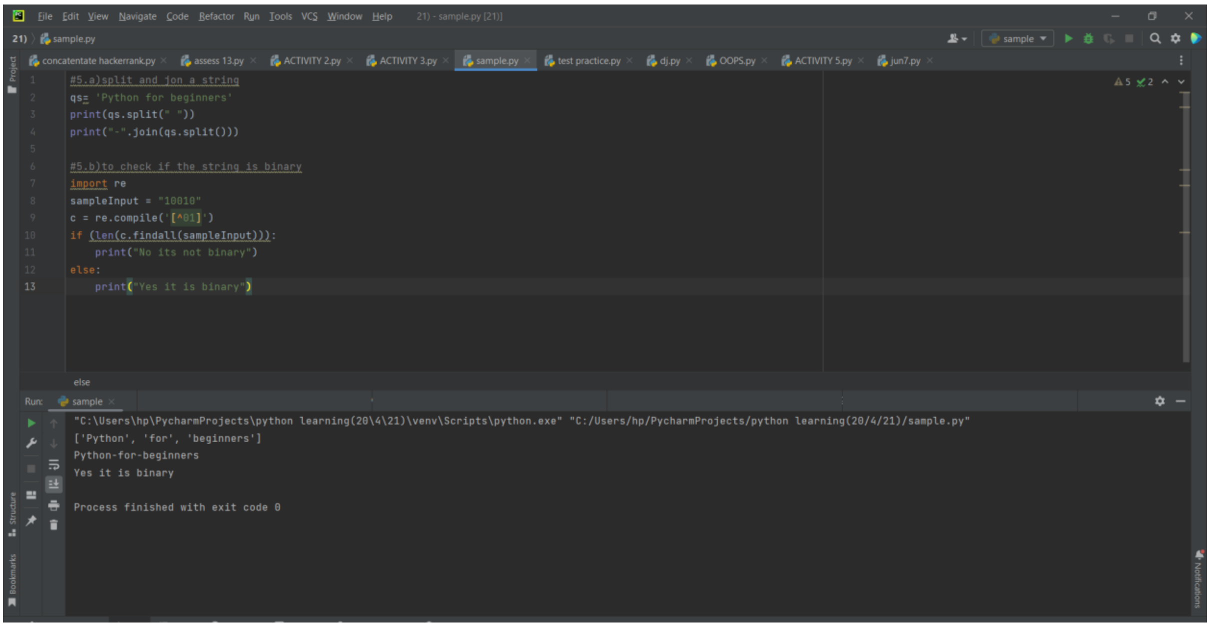


```
#4.to print negative numbers in a range
start, end = -4, 0
for num in range(start, end + 1):
    if num < 0:
        print(num, end=" ")
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/sample.py"
-4 -3 -2 -1
Process finished with exit code 0
```

```
#5.a)split and jon a string
qs= 'Python for beginners'
print(qs.split(" "))
print("-".join(qs.split()))
```

```
#5.b)to check if the string is binary
import re
sampleInput = "10010"
c = re.compile('[^01]')
if (len(c.findall(sampleInput))):
    print("No its not binary")
else:
    print("Yes it is binary")
```

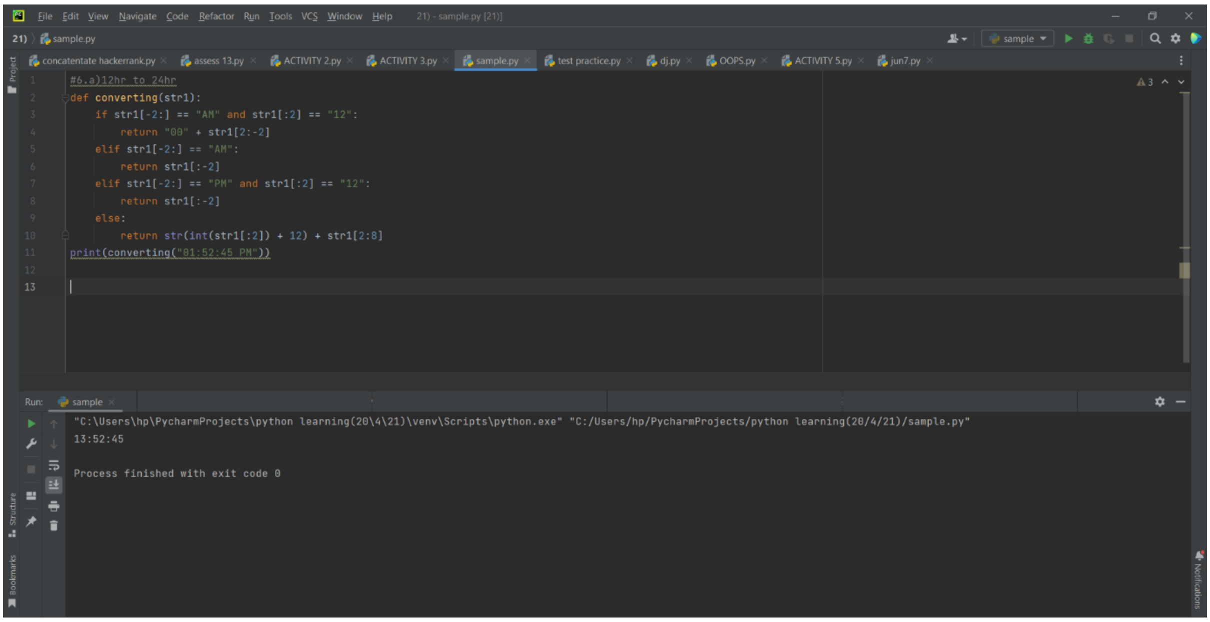


```
#5.a)split and jon a string
qs= 'Python for beginners'
print(qs.split(" "))
print("-".join(qs.split()))

#5.b)to check if the string is binary
import re
sampleInput = "10010"
c = re.compile('[^01]')
if (len(c.findall(sampleInput))):
    print("No its not binary")
else:
    print("Yes it is binary")
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/sample.py"
['Python', 'for', 'beginners']
Python-for-beginners
Yes it is binary

Process finished with exit code 0
```

#6.a)12hr to 24hrs:

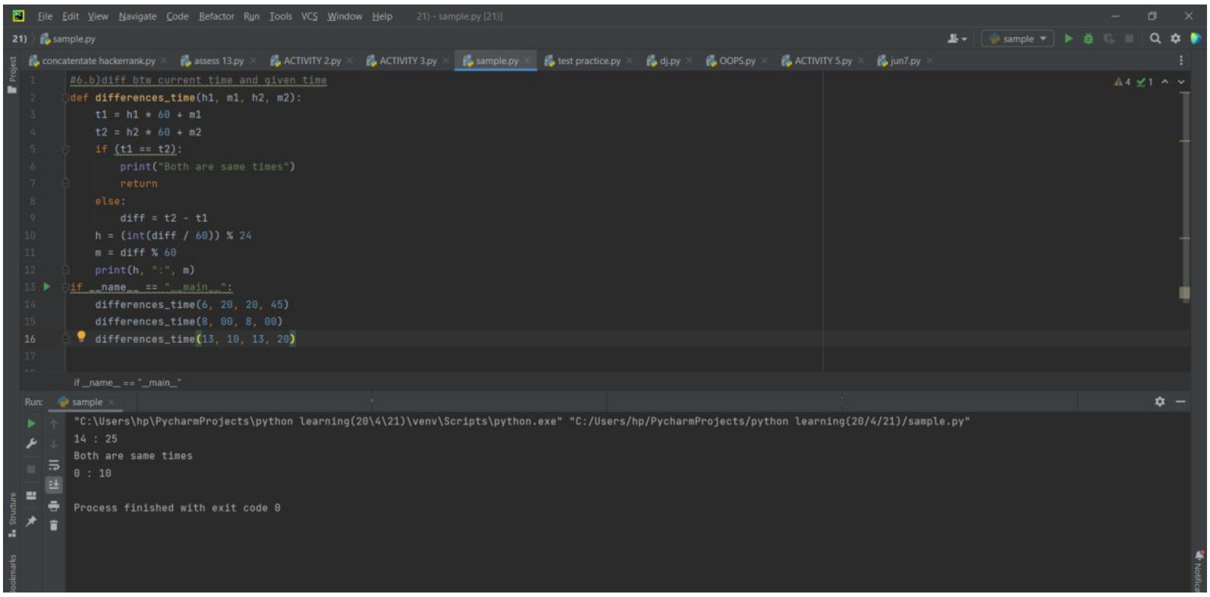
```
def converting(str1):
    if str1[-2:] == "AM" and str1[:2] == "12":
        return "00" + str1[2:-2]
    elif str1[-2:] == "AM":
        return str1[:-2]
    elif str1[-2:] == "PM" and str1[:2] == "12":
        return str1[:-2]
    else:
        return str(int(str1[:2]) + 12) + str1[2:8]
print(converting("01:52:45 PM"))
```



```python
#6.a)12hr to 24hr
def converting(str1):
    if str1[-2:] == "AM" and str1[:2] == "12":
        return "00" + str1[2:-2]
    elif str1[-2:] == "AM":
        return str1[:-2]
    elif str1[-2:] == "PM" and str1[:2] == "12":
        return str1[:-2]
    else:
        return str(int(str1[:2]) + 12) + str1[2:8]
print(converting("01:52:45 PM"))
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/sample.py"
13:52:45

Process finished with exit code 0
```

```
#6.b)diff btw current time and given time
def differences_time(h1, m1, h2, m2):
    t1 = h1 * 60 + m1
    t2 = h2 * 60 + m2
    if (t1 == t2):
        print("Both are same times")
        return
    else:
        diff = t2 - t1
    h = (int(diff / 60)) % 24
    m = diff % 60
    print(h, ":", m)
if __name__ == "__main__":
    differences_time(6, 20, 20, 45)
    differences_time(8, 00, 8, 50)
    differences_time(13, 10, 13, 20)
```



```python
#6.b)diff btw current time and given time
def differences_time(h1, m1, h2, m2):
    t1 = h1 * 60 + m1
    t2 = h2 * 60 + m2
    if (t1 == t2):
        print("Both are same times")
        return
    else:
        diff = t2 - t1
    h = (int(diff / 60)) % 24
    m = diff % 60
    print(h, ":", m)
if __name__ == "__main__":
    differences_time(6, 20, 20, 45)
    differences_time(8, 00, 8, 00)
    differences_time(13, 10, 13, 20)
```

```
"C:\Users\hp\PycharmProjects\python learning(20\4\21)\venv\Scripts\python.exe" "C:/Users/hp/PycharmProjects/python learning(20/4/21)/sample.py"
14 : 25
Both are same times
0 : 10

Process finished with exit code 0
```