

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

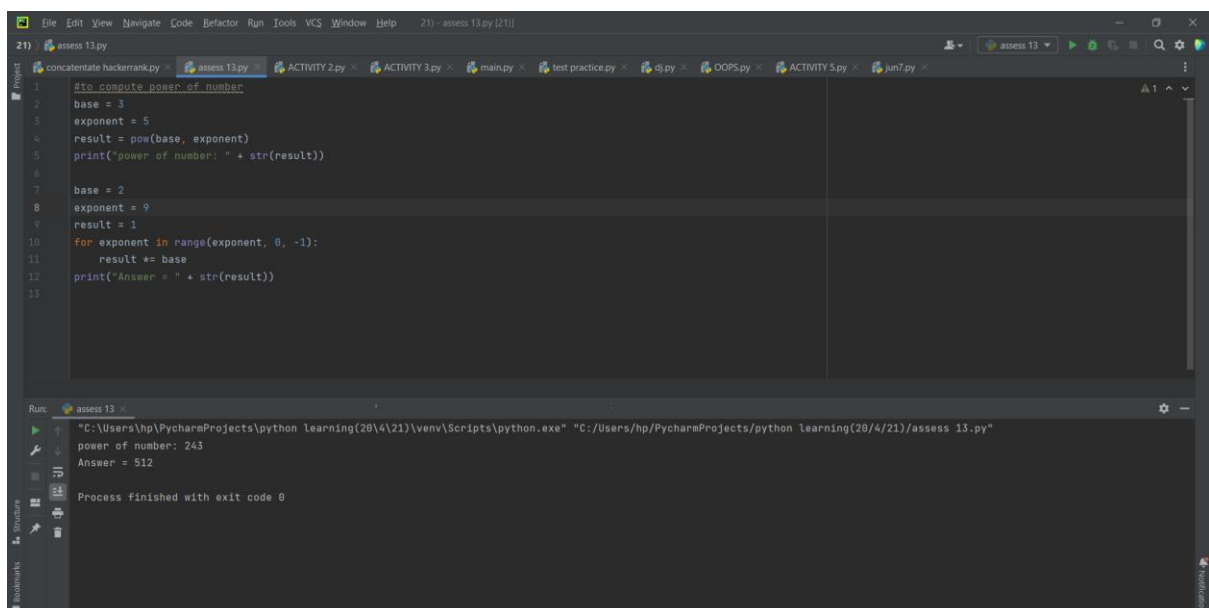
exponent = 9

result = 1

for exponent in range(exponent, 0, -1):

 result *= base

print("Answer = " + str(result))



The screenshot shows a PyCharm IDE window with a Python script named 'assess 13.py'. The script contains two methods for calculating powers. The first method uses the built-in 'pow' function to calculate 3 to the power of 5. The second method uses a 'for' loop to calculate 2 to the power of 9. The 'Run' console at the bottom shows the output of the script: 'power of number: 243' and 'Answer = 512'. The process finished with exit code 0.

```
1 #to compute power of number
2 base = 3
3 exponent = 5
4 result = pow(base, exponent)
5 print("power of number: " + str(result))
6
7 base = 2
8 exponent = 9
9 result = 1
10 for exponent in range(exponent, 0, -1):
11     result *= base
12 print("Answer = " + str(result))
13
```

Run: assess 13.py
"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
```

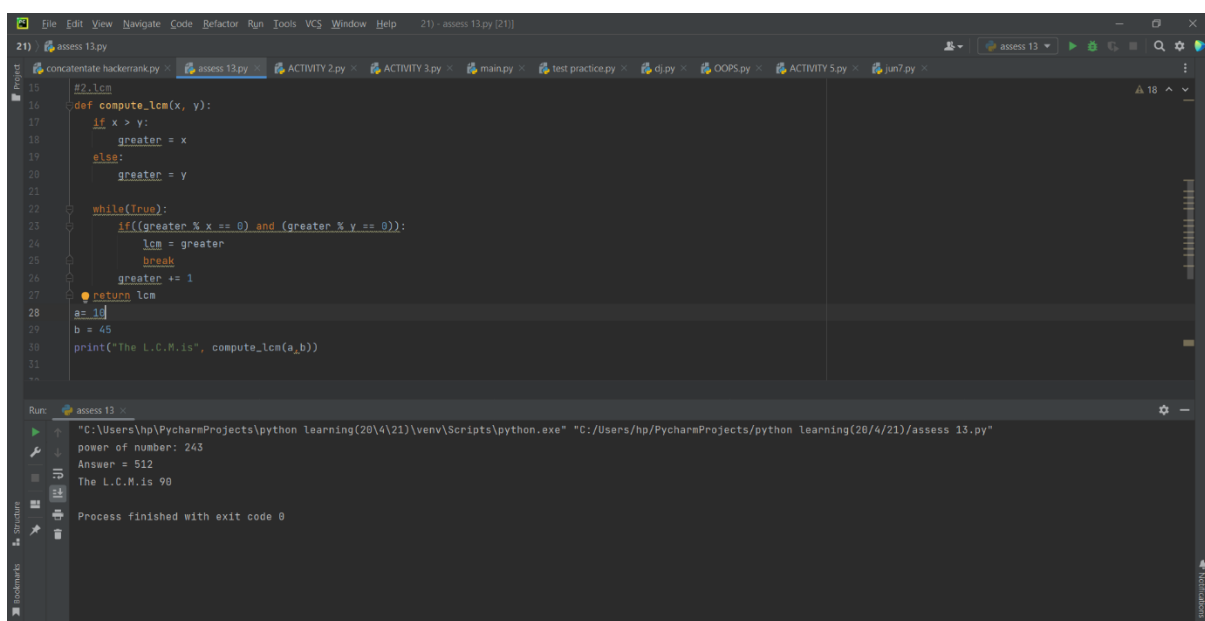
```
        greater += 1
```

```
    return lcm
```

```
a= 10
```

```
b = 45
```

```
print("The L.C.M.is", compute_lcm(a,b))
```



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help 21 - assess 13.py [21]
21) assess 13.py concatenate hackerrank.py assess 13.py ACTIVITY 2.py ACTIVITY 3.py main.py test practice.py dj.py OOPS.py ACTIVITY 5.py jn7.py
15 #2.lcm
16 def compute_lcm(x, y):
17     if x > y:
18         greater = x
19     else:
20         greater = y
21
22     while(True):
23         if((greater % x == 0) and (greater % y == 0)):
24             lcm = greater
25             break
26         greater += 1
27     return lcm
28 a= 10
29 b = 45
30 print("The L.C.M.is", compute_lcm(a,b))
31
Run: assess 13
"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
```

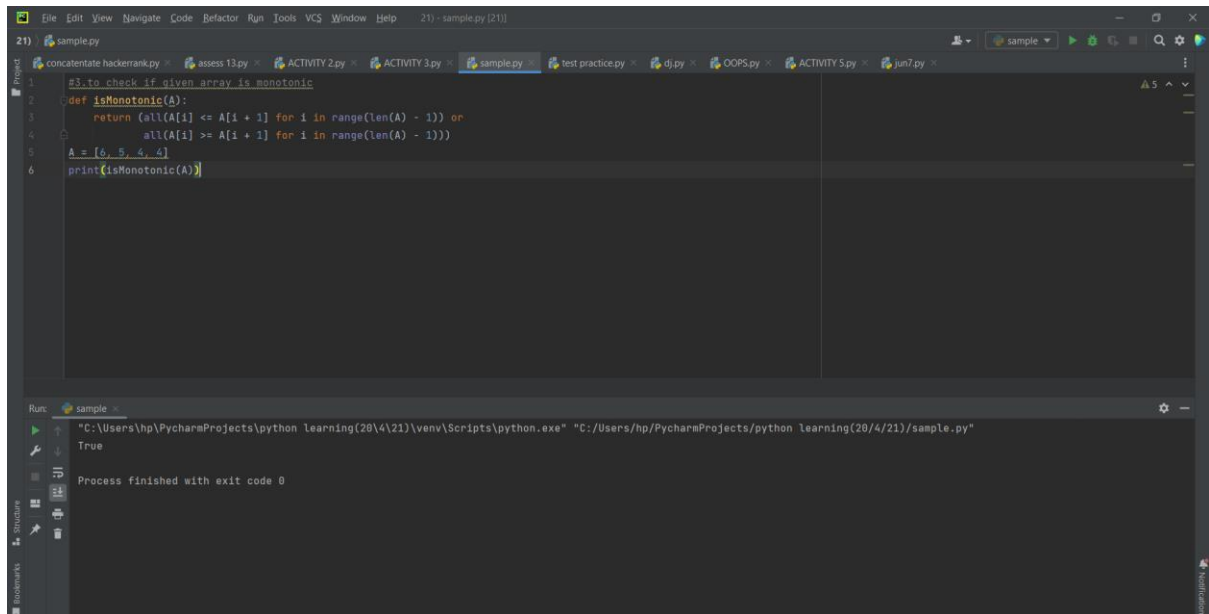
#3.to check if given array is monotonic

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))



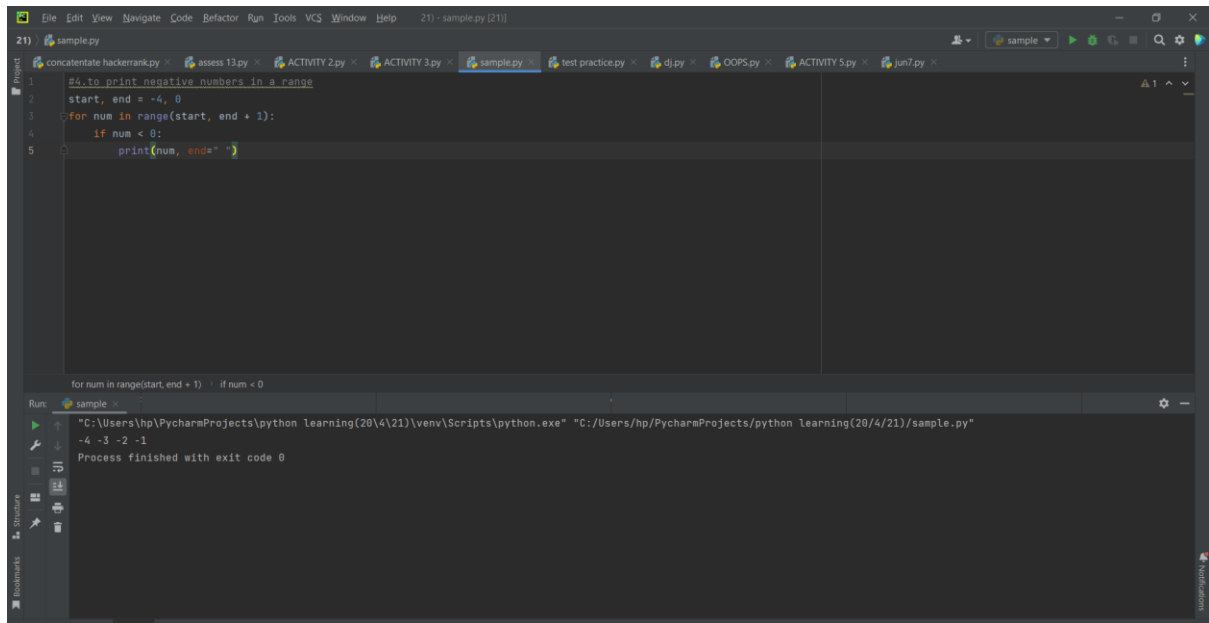
The screenshot shows a PyCharm IDE window with a Python file named 'sample.py'. The code in the file is as follows:

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

The 'Run' tab at the bottom shows the execution output:

```
Run: sample
"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=" ")
```

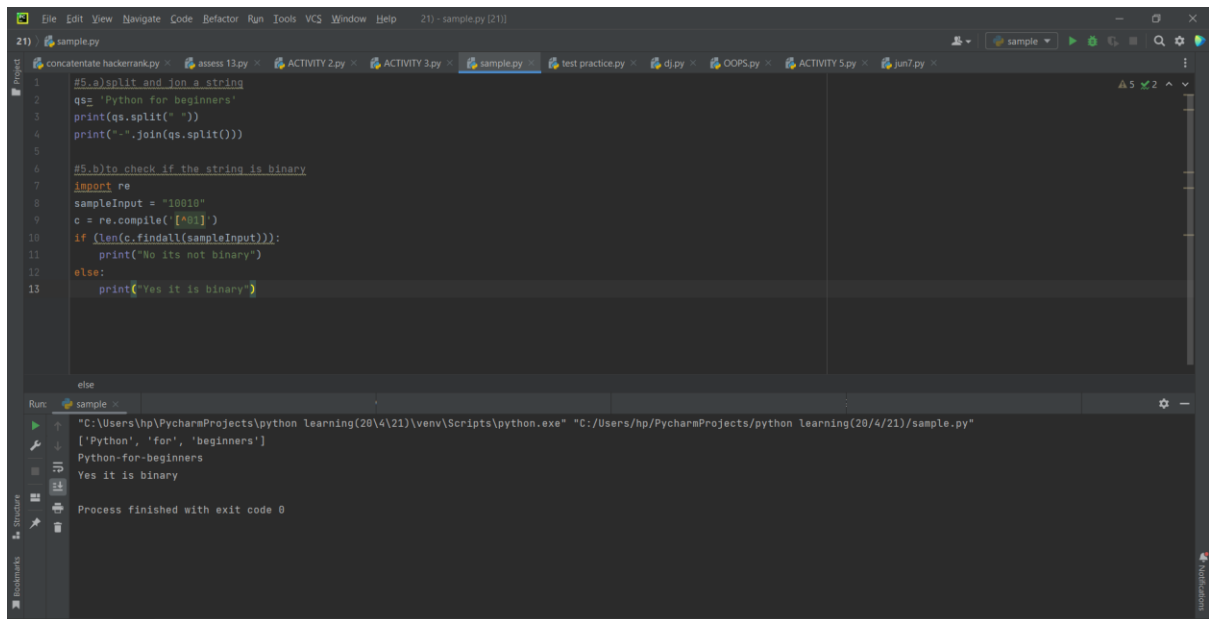
A screenshot of the PyCharm IDE interface. The main editor window shows a Python script with the following code:

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

The script is named 'sample.py'. Below the editor, the 'Run' console shows the output: '-4 -3 -2 -1' followed by 'Process finished with exit code 0'. The console also displays the command used to run the script: '"C:/Users/hp/PycharmProjects/python_learning(20/4/21)/venv/Scripts/python.exe" "C:/Users/hp/PycharmProjects/python_learning(20/4/21)/sample.py"'. The PyCharm interface includes a top menu bar, a toolbar, and a sidebar on the left with icons for Project, Structure, and Run/Debug.

```
#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")
```



#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]

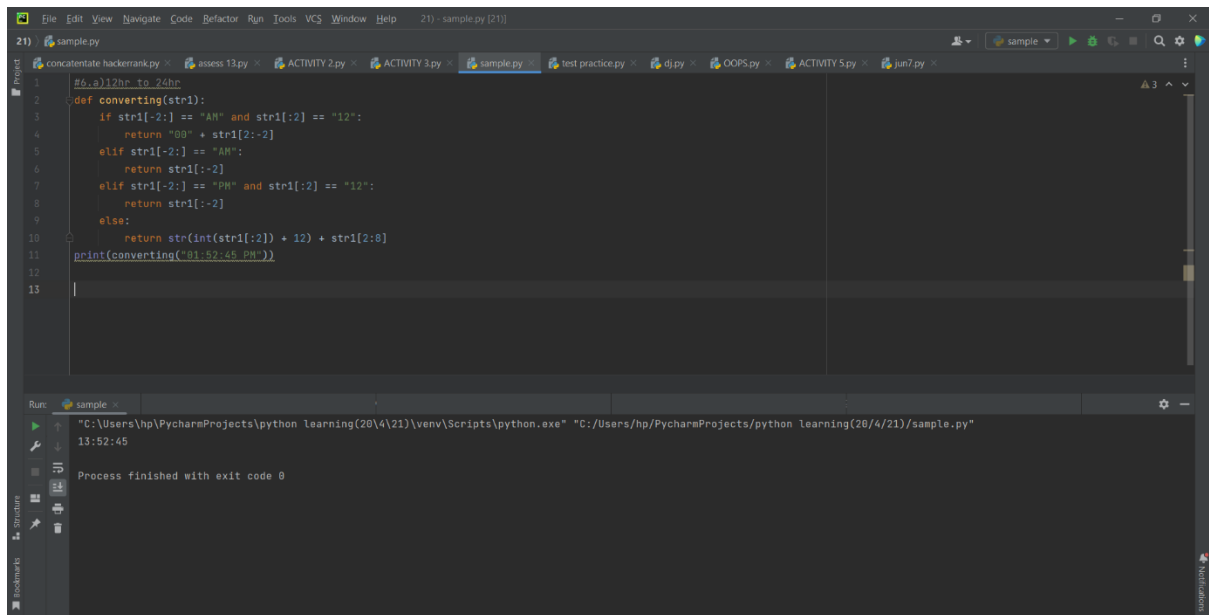
elif str1[-2:] == "PM" and str1[:2] == "12":

return str1[:-2]

else:

return str(int(str1[:2]) + 12) + str1[2:8]

print(converting("01:52:45 PM"))



#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)

The image shows a PyCharm IDE window with a Python script named `sample.py` open. The script defines a function `differences_time(h1, m1, h2, m2)` that calculates the difference between two times. The function returns `"Both are same times"` if the times are equal, and otherwise returns the difference in hours (`h`) and minutes (`m`). The script also includes a `__main__` block that calls the function with three sets of input values: `(6, 20, 20, 45)`, `(8, 00, 8, 00)`, and `(13, 10, 13, 20)`.

```
1 #6.b)diff btw current time and given time
2 def differences_time(h1, m1, h2, m2):
3     t1 = h1 * 60 + m1
4     t2 = h2 * 60 + m2
5     if (t1 == t2):
6         print("Both are same times")
7         return
8     else:
9         diff = t2 - t1
10        h = (int(diff / 60)) % 24
11        m = diff % 60
12        print(h, ":", m)
13 if __name__ == "__main__":
14     differences_time(6, 20, 20, 45)
15     differences_time(8, 00, 8, 00)
16     differences_time(13, 10, 13, 20)
17
18 if __name__ == "__main__":
```

The Run console at the bottom shows the output of the script:

```
sample
"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
```