**Assignment 2**

The values of Boolean datatype are True and False and we write T for true and F for false.

**2.**

AND, OR and NOT

**3.**

Truth tables

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P and Q** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P or Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **P** | **Q** | **P ^ Q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  |  |
| --- | --- |
| **P** | **not P** |
| 0 | 1 |
| 1 | 0 |

**4.**

(5>4) and (3==5) F   
not (5>4) T   
(5>4) or (3==5) T   
not ((5>4) or (3==5)) F   
(True and True) and (True==False) F   
(not False) or (not True) T

**5.**

==   
<   
>   
<=   
>=   
!=

**6.**

‘=’ sign is used to assign a value and ‘==’ is used to compare two values.   
Ex- a=25   
      if a==25: print(‘equal’)

**7.**

spam = 0   
if spam == 10:   
print(‘eggs’)   
if spam > 5:   
print(‘bacon’)   
else:    
print(‘ham’)   
print(‘spam’)   
print(‘spam’)

**8.**

If spam == 1:   
print(‘Hello’)   
if spam == 2:   
print(‘Howdy’)   
else:   
print(‘Greetings!’)

**9.**

Ctrl + C

**10.**

break statement will terminate the loop but continue statement only skips one iteration of the loop.

**11.**

Intuitively, range makes a lot of sense. It’s used as follows: range([start], stop[, step]).   
start is optional. When not provided, it defaults to 0. It is the starting number of the sequence.   
stop is mandatory. range will generate numbers up to but not including the value of stop.   
step is optional. When not provided, it defaults to 1. It is the difference between two elements in the sequence. It cannot be 0; in this case, a ValueError would be raised.   
So range(10), range(0,10) and range(0,10,1) give the same output but due to different methods

**12.**

for i in range(1,11):   
print(i)   
   
i=1   
while(i<11)   
print(i)   
i = i + 1

**13.**

import spam as sp   
sp.bacon()