TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN MÔN DISCRETE STRUCTURES

ESSAY

Người hướng dẫn: **GV NGUYỄN QUỐC BÌNH**

Người thực hiện: **PHẠM PHƯỚC TẤN – 520H0418**

Lớp : **20H50204**

Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN MÔN DISCRETE STRUCTURES

ESSAY

Người hướng dẫn: **GV NGUYỄN QUỐC BÌNH**
Người thực hiện: **PHẠM PHƯỚC TẤN**
Lớp **: 20H50204**
Khoá **: 24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

LỜI CẢM ƠN

I’m really grateful for instruction and explanation of Lecturer who name is Nguyễn Quốc Bình about Essay of Discrete Structures.Therefore, I could complete my essay turning into better.

TÓM TẮT

In the essay, The trouble would swing about the theory of Reverse Polish and basic logic and to be solved then we will handle on input from Infix including some logical operators and Alphabet characters from A to Z and then output of Infix which is Postfix. That mean the operators will follow their operands.For example, the logical operators is 8 - 9, one would write 8 9 - instead of 8 - 9 so that’s Reverse Polish. From output of reverse polish then that is Postfix, we would check be precedence from Postfix following list Alphabets and operators.After that, we will solve between two alphabet characters and an operator that calculate it in shape of True and False, output of it is True or False. After calculated, we will add from output to the truth table.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

f	*Tần số của dòng điện và điện áp (Hz)*
p	*Mật độ điện tích khối (C/m3)*

CÁC CHỮ VIẾT TẮT

CSTD	Công suất tác dụng
MF	Máy phát điện
BER	Tỷ lệ bít lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

CHAPTER 1 – THE FIRST

1.1 Introduce my group

My essay is only me and I wrote code by my side.

1.2 Introduce what other chapters do

Second chapter includes the theory of Reverse Polish and Basic logic.

Third chapter would explain my program by doing step by step each function.

Fourth chapter is necessary my experimental result on 5 testcases.

Fifth chapter lists all the references.

CHAPTER 2 – THE SECOND

2.1 Theory of Reverse Polish

Reverse Polish Notation is a mathematical notation so it's necessary quite well in a stack implement. Besides that, the operators depended by their operands and logic operators must to be precedence from not, and , or , Implication and Bi implication. However, Reverse Polish don't follow the open and close parenthesis.

For instance,

In ordinary (or also call infix) an logical operators: Q & (P | R)

After reverse polish notation, the same logical operators would be written as:QPR|&

2.2 Theory of Basic Logic

2.2.1 Logical NOT Operation

Logical NOT operation is the negation of a given character.For example, an original character is R so the negation of it is ~R that mean if R is true so ~R is false, otherwise

2.2.2 Logical AND Operation

Logical AND operation is the same boolean AND operation or a statement of the programming language.For instance, two characters is R and Q so it is true when both of it is true and otherwise is false.

2.2.3 Logical OR Operation

Logical OR operation is a set of operands and it's true when one or two character is true , it's false when both it it is false.For example, two characters is R which is true and Q which is False, R or Q that is True. Otherwise, R is false and Q is false and then R or Q is false.

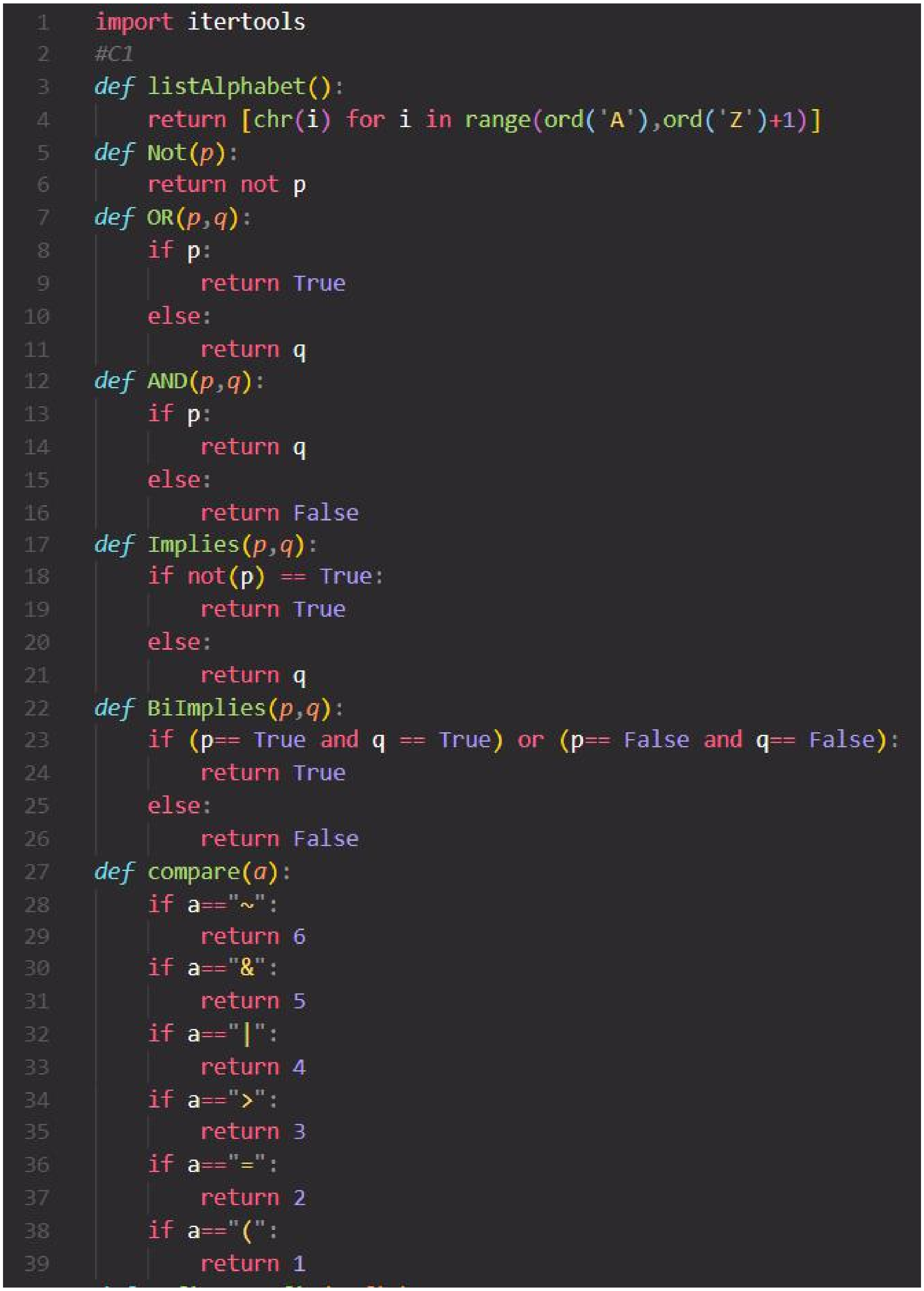
2.2.4 Logical Implication

Logical implication is a relationship between two statement and the second is the consequence of the first and implication is like if then statement so it signified the first statement -> the second statement. For example, two characters is R and Q, R implies Q (R->Q) so it's false when R is true and Q is false. Otherwise is true.

2.2.5 Logical Bi-Implication

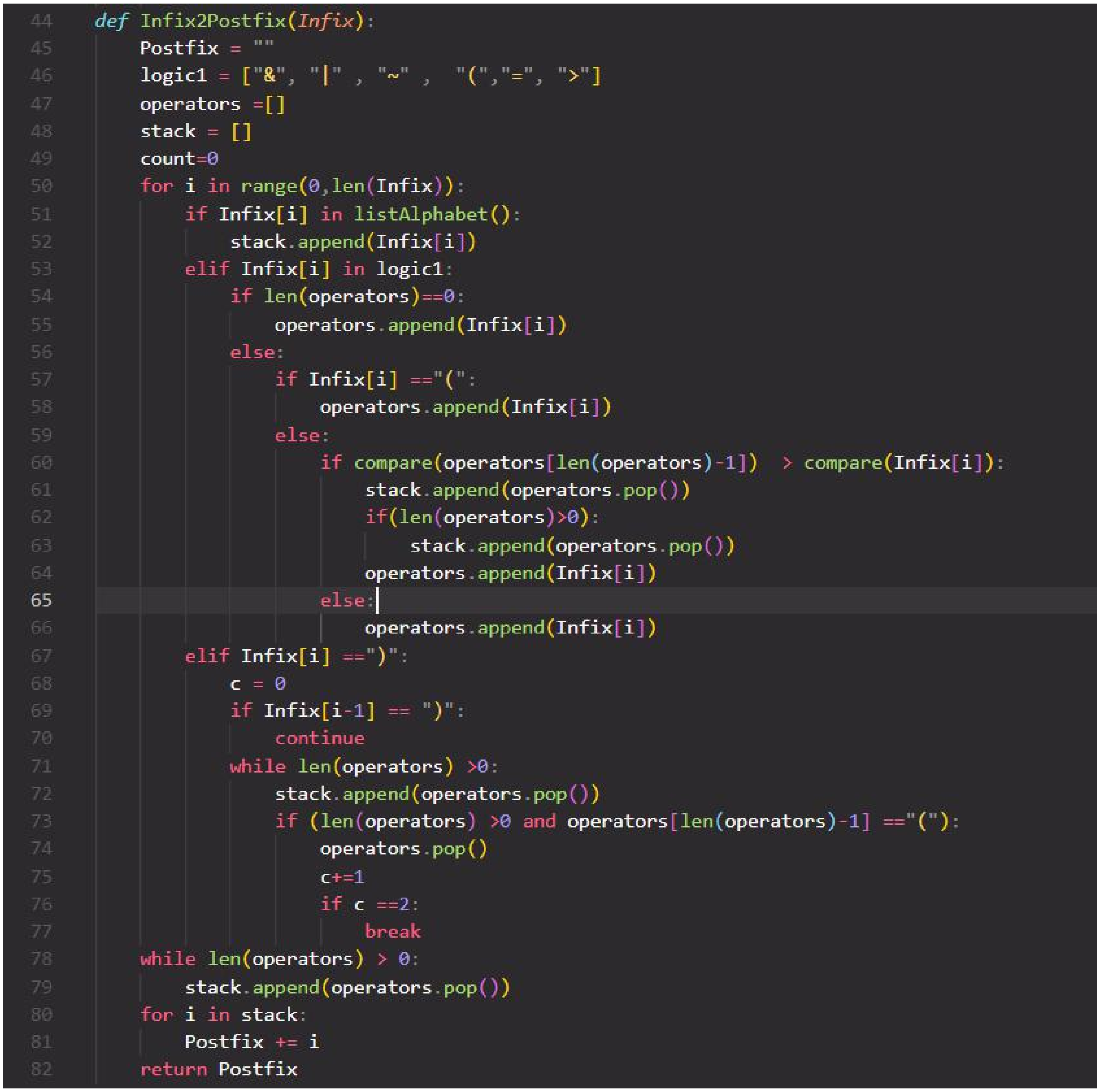
Logical bi-implication is like the if and only if statement and it signified the first statement <-> the second statement. Besides that, two statement dependent on each other. For example, two characters is R and Q, R bi-implies Q (R<->Q) so it's true when R and Q is the same operation.Otherwise is false.

CHAPTER 3 – THE THIRD



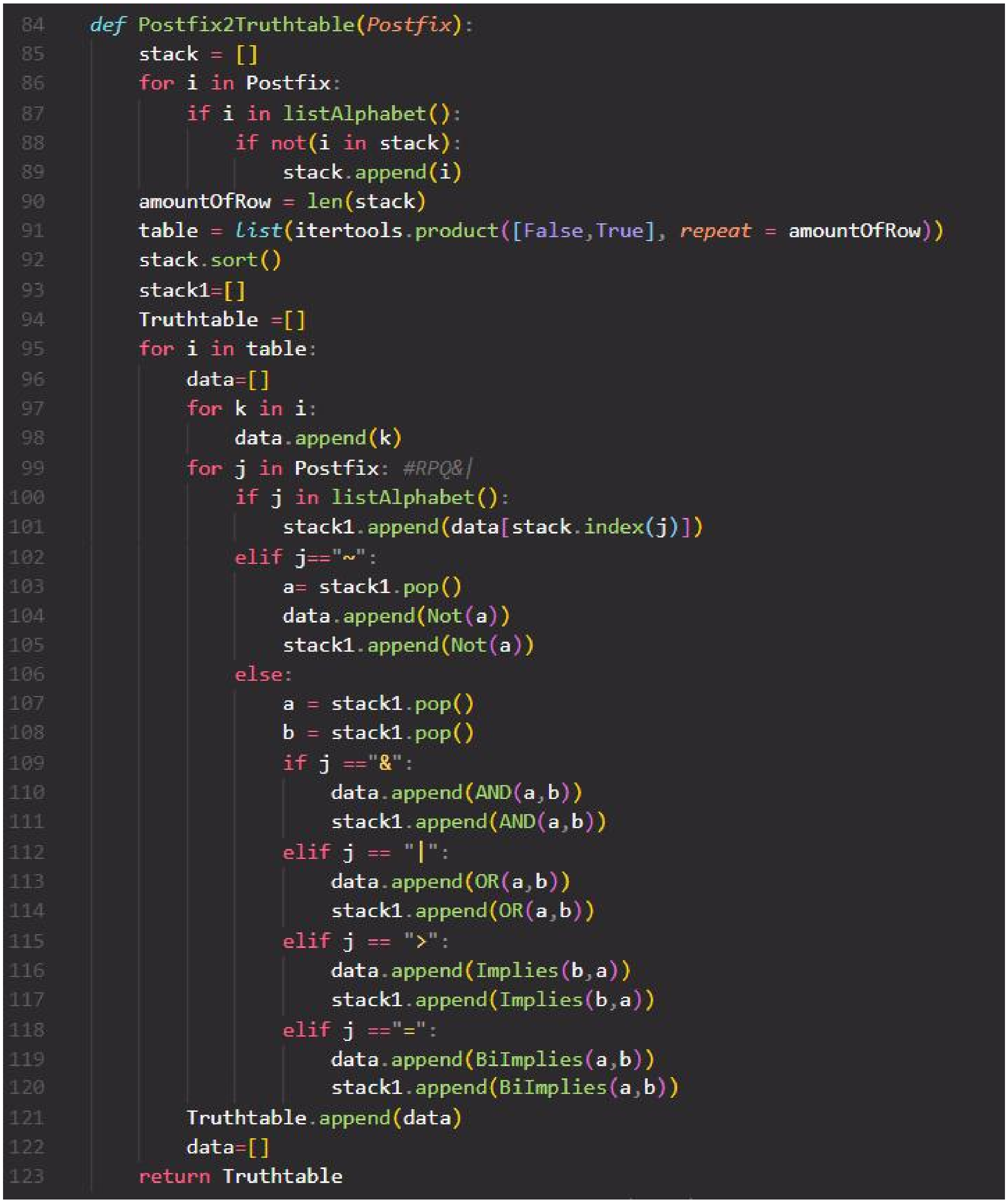
```python
import itertools
#C1
def listAlphabet():
    return [chr(i) for i in range(ord('A'),ord('Z')+1)]
def Not(p):
    return not p
def OR(p,q):
    if p:
        return True
    else:
        return q
def AND(p,q):
    if p:
        return q
    else:
        return False
def Implies(p,q):
    if not(p) == True:
        return True
    else:
        return q
def BiImplies(p,q):
    if (p== True and q == True) or (p== False and q== False):
        return True
    else:
        return False
def compare(a):
    if a=="~":
        return 6
    if a=="&":
        return 5
    if a=="|":
        return 4
    if a==">":
        return 3
    if a=="=":
        return 2
    if a=="(":
        return 1
```

Picture 1: The functions



```python
def Infix2Postfix(Infix):
    Postfix = ""
    logic1 = ["&", "|" , "~" ,  "(","=", ">"]
    operators =[]
    stack = []
    count=0
    for i in range(0,len(Infix)):
        if Infix[i] in listAlphabet():
            stack.append(Infix[i])
        elif Infix[i] in logic1:
            if len(operators)==0:
                operators.append(Infix[i])
            else:
                if Infix[i] =="(":
                    operators.append(Infix[i])
                else:
                    if compare(operators[len(operators)-1])  > compare(Infix[i]):
                        stack.append(operators.pop())
                        if(len(operators)>0):
                            stack.append(operators.pop())
                        operators.append(Infix[i])
                    else:
                        operators.append(Infix[i])
        elif Infix[i] ==")":
            c = 0
            if Infix[i-1] == ")":
                continue
            while len(operators) >0:
                stack.append(operators.pop())
                if (len(operators) >0 and operators[len(operators)-1] =="("):
                    operators.pop()
                    c+=1
                    if c ==2:
                        break
    while len(operators) > 0:
        stack.append(operators.pop())
    for i in stack:
        Postfix += i
    return Postfix
```

Picture 2: Function def Infix2Postfix(Infix)



```python
def Postfix2Truthtable(Postfix):
    stack = []
    for i in Postfix:
        if i in listAlphabet():
            if not(i in stack):
                stack.append(i)
    amountOfRow = len(stack)
    table = list(itertools.product([False,True], repeat = amountOfRow))
    stack.sort()
    stack1=[]
    Truthtable =[]
    for i in table:
        data=[]
        for k in i:
            data.append(k)
        for j in Postfix: #RPQ&|
            if j in listAlphabet():
                stack1.append(data[stack.index(j)])
            elif j=="~":
                a= stack1.pop()
                data.append(Not(a))
                stack1.append(Not(a))
            else:
                a = stack1.pop()
                b = stack1.pop()
                if j =="&":
                    data.append(AND(a,b))
                    stack1.append(AND(a,b))
                elif j == "|":
                    data.append(OR(a,b))
                    stack1.append(OR(a,b))
                elif j == ">":
                    data.append(Implies(b,a))
                    stack1.append(Implies(b,a))
                elif j =="=":
                    data.append(BiImplies(a,b))
                    stack1.append(BiImplies(a,b))
        Truthtable.append(data)
        data=[]
    return Truthtable
```

Picture 3: Function def Postfix2Truthtable(Postfix)

3.1 Testcase 1: R|(P&Q)

3.1.1 Function 1

At function 1, input is Infix which is with a string of logical operators and input of test case 1 is "R|(P&Q)".

The first, I will create a string with output which contains logical operators after using the reverse polish notation methodology and a characters array from the operators like top and then create a operators stack that contain logical operators including AND, OR, NOT, Implication, Bi-implication, a stack array that use containing after calculate the String Infix and final I create a variable which is count =0.

Using loop for browsing from number 0 to the length of a string Infix, because Infix has seven element so will loop seven times:

-The first element is i =0 with Infix[0] which is "R" and "R" is a character in listAlphabet() as "R" is a alphabet character that I create a function that name is listAlphabet() at a picture 1.so "R" have to true with if condition so I will add it in stack array via syntax which is stack.append(Infix[i]) meaning add the Infix[0] data in the last stack array. Final, the stack array has ["R"] element and finish loop with i =0.

-The second element is i =1 with Infix[1] which is "|" and it is true at the else condition that's in line 53 because "|" isn't the alphabet characters, continue it meet the if condition at line 54, which if the length of operators array is empty then it's added Infix[i] in the last operators array, as the length of operators array is empty so it's added Infix[1] in the last operators. Final, the operators array has ["|"] and stack =["R"] and finish with i = 1.

-The third element is i = 2 with Infix[2] which is "(" , it will meet the else condition at line 53 because it's the logical operators and continue, it will meet the else condition at line 56 because the operators array is currently the length equal 1.Continue, it will meet the if condition at line 57 as Infix[i] == "(" so it's going this the if condition at line 57 and in it will perform adding an

Infix[2] element in the last operators array so the operator array has ["|","("] and stack array has ["R"] and finish at i =2.

-The fourth element is i = 3 with Infix[3] which is "P" so will join the if condition at line 51 and adding Infix[3] in the last stack array so the stack array has ["R","P"], the operator array has ["|","("] and finish at i = 3.

-The fifth element is i = 4 with Infix[4] which is "&" so will join the else condition at line 53 and continue joining the else condition at line 59 because Infix[4] isn't "(" so join it. Continue, meeting the if condition and compare the precedence of two logic operators and compare two last element of the operators array with Infix[i], the last element of an operators array which is "(" and comparing to Infix[4], if the last operators array has the precedence higher Infix[4] then adding the last operators array in the last stack array, at the same time, deleting the last element of an operators array and otherwise, I will add the Infix[4] element in the operators array, in this top situation, an Infix[4] has the precedence higher the last element operators array because the function is checked the priority from top to bottom which following "~,&, |, >, =, (" (follow this table in document essay, so Infix[4] added in last operators array. The final, the operators array has ["|", "(", "&"], stack =["R","P"] and finish i =4.

-The sixth element is i = 5 with Infix[5] which is "Q" so will join the if condition at line 51 and adding Infix[5] in the last stack array so the stack array has ["R","P","Q"], the operator array has ["|","(","&"] and finish at i = 5

-The seventh element is i = 6 with Infix[6] which is ")" so it will meet the if condition at line 67 and it start in the if condition then I create a variable that name is i =0, that check delete how many the "(" operation of the last operators array. And the if condition at line 69 that check in front of the Infix String element is equal ")" or not, if carry on continue for loop, because

Infix[6-1] = Infix[5] =Q so it's not in this if condition and it will meet the while loop at line 71 , if the length of operators array not empty then pop() each element of the operators array that add in last stack array as the operators array is not empty so add the last element of operators array which is "&" in the last stack array at the same time, deleting the last element just adding of the operators array via syntax stack.pop() and the currently stack array is ["R","P" , "Q" , "&"] and operators =["|" , "("]. After that, continue it meet the if condition at line 73 if the operators array is not empty and a last element of the operators array equal "(" then go in the if condition, because the operators array is not empty and a last element of the operators array is "(" so go in the if condition and in the if condition carry on deleting a last element of the operators via syntax operators.pop(). After that, deleting a last element of the operators array and after delete, the operators array is remaining ["|"] and increasing the C variable 1 unit, the C variable is 1 number so it's not going in the if condition and the while loop keep going necessary because the length of operators array is not empty so continue. Adding and deleting a last element of the operators array in the last stack array, the last element is "|" and adding in the last stack array. Presently, the stack array has "R", "P" , "Q" , "&" , "|" and the operators array is empty so it's not going in the if condition and final it will finish the while loop because the length of the operators is empty and finish at the loop with i =6.

-So the infix string browsed to the last string therefore finish the loop browse the Infix string. In short, the stack array has "R", "P" , "Q" , "&" , "|" and the operators is empty . After that, it will meet the while loop to check the length of operators array which is not empty then keep going adding and deleting a last element of the operators array via syntax operators.pop() and adding a last element of the stack array, but the length of operators array is empty so it's not carrying on the while loop.

Continue, I will add each element of the stack array in the Postfix String via syntax using the for loop to browse all element of the stack array at line 80 and adding a element of the stack array in the Postfix string via syntax Postfix = Postfix + each element of the stack array because it from the array to the string so it must to use +=, if not,then in the string will has "".

After adding the data from the stack array to the Postfix string then I will return the Postfix String at the last function.

3.1.2 Function 2

After solve function 1 so input of Postfix is RPQ&|.I will create a stack array that contains the alphabet characters.I will use loop For in Postfix browsing each element in Postfix,if element i of an element Postfix which is alphabet character and not in stack so add an element i in last stack array. The first with i = "R" which is in list alphabet characters and not in stack array so add element i in last stack array therefore, stack array has "R" character, with i ="P" which is alphabet character and not in stack array so add i ="P" in stack array so stack array element has "R","P".Continue i = "Q" which is alphabet character and not in stack array so add i ="Q" in stack array so stack array element has "R","P","Q".Continue i ="&" which is logical operator and not an alphabet character so not add in stack array. Continue i ="|" which is logical operator and not an alphabet character so not add in stack array. Final, We have the stack array which contain "R","P","Q".

Continue, I will create an integer data type that solve length of stack array, with len(stack) at line 90 so an amount of row is 3 because the stack array has three elements.

I will create a table which contain true and false of the alphabet characters of stack array and with amount of column which is the amountOfRow variable that solved equal 3,so we have a table with three columns.

And then, I will sort stack array in alphabetical order from A to Z so stack array will sort and the result is "P", "Q", "R". I will create stack1 array which contains the data which is true, false of the alphabet of stack. final I create a Truthtable array that save the calculates in table.

I will use the for loop that browse each row of the table with for i in table, and then I will create a data array that contains a data which is true or false of each element of i , continue, I use the for loop to browse each element of i and then in it so I take each element of i which is k adding k in the last of data array.

The first row of table with i including False False False and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [False] and finishing the first loop so continue, the second loop with k = false so I add it in the last data so the data array has [False,False] and finishing the second loop. Continue the third loop with k = false so I add it in the last data so the data array has [False,False,False] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is False because the data array has [False,False,False] after data[2] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is False because the data array has [False,False,False] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,False] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is False because the data array has [False,False,False] after data[1] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,False,False] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is False of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is False of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [False,False,False,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [False,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal False of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is False and I will add it on the last element of data array so the data array has [False,False,False,False,False] after that I still add OR(a,b) in the last stack1 so the stack1 is [False] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [False,False,False,False,False] and after that I will reset the data array.

The second row of table with i including False False True and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [False] and finishing the first loop so continue, the second loop with k = false so I add it in the last data so the data array has [False,False] and finishing the second loop. Continue the third loop with k = true so I add it in the last data so the data array has [False,False,True] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is True because the data array has [False,False,True] after data[2] is

True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is False because the data array has [False,False,True] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True,False] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is False because the data array has [False,False,True] after data[1] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True,False,False] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is False of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is False of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [False,False,True,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [True,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal True of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is True and I will add it on the last element of data array so the data array has

[False,False,True,False,True] after that I still add OR(a,b) in the last stack1 so the stack1 is [True] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [False,False,True,False,True] and after that I will reset the data array.

The third row of table with i including False True False and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [False] and finishing the first loop so continue, the second loop with k = true so I add it in the last data so the data array has [False,True] and finishing the second loop. Continue the third loop with k = false so I add it in the last data so the data array has [False,True,False] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is False because the data array has [False,True,False] after data[2] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is False because the data array has [False,True,False] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,False] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is True because the data array has [False,True,False] after data[1] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,False,True] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is True of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is False of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [False,True,False,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [False,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal False of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is False and I will add it on the last element of data array so the data array has [False,True,False,False,False] after that I still add OR(a,b) in the last stack1 so the stack1 is [False] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [False,True,False,False,False] and after that I will reset the data array.

The fourth row of table with i including False True True and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [False] and finishing the first loop so continue, the second loop with k = true so I add it in the last data so the data array has [False,True] and finishing the second loop. Continue the third loop with k = true so I add it in the last data so the data array has [False,True,True] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order

3 and then we take a data of j from the data array via syntax data[2] so it is True because the data array has [False,True,True] after data[2] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is False because the data array has [False,True,True] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True,False] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is True because the data array has [False,True,True] after data[1] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True ,False,True] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is True of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is False of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [False,True,True ,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [True ,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal True of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is True and I will add it on the last element of data array so the data array has [False,True,True ,False,True] after that I still add OR(a,b) in the last stack1 so the stack1 is [True] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [False,True,True ,False,True] and after that I will reset the data array.

The fifth row of table with i including True False False and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [True] and finishing the first loop so continue, the second loop with k = False so I add it in the last data so the data array has [True,False] and finishing the second loop. Continue the third loop with k = False so I add it in the last data so the data array has [True,False ,False] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is False because the data array has [True,False ,False] after data[2] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is True because the data array has [True,False ,False] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,True] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is False because the data array has [True,False ,False] after data[1] is

True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,True,False] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is False of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is True of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [True,False,False,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [False,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal False of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is False and I will add it on the last element of data array so the data array has [True ,False,False,False,False] after that I still add OR(a,b) in the last stack1 so the stack1 is [False] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [True ,False,False,False,False] and after that I will reset the data array.

The sixth row of table with i including True False True and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [True] and finishing the first loop so continue, the second loop with k = False so I add it in the last data so the data array has [True,False] and finishing the second loop. Continue the third loop with k = True so I add it in the last data so the data array has [True,False ,True] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is True because the data array has [True,False ,True] after data[2] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True] of R and finish at the first loop is j.

-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is True because the data array has [True,False ,True] after data[0] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,True] of R,P and finish at the second loop is j.

-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is False because the data array has [True,False ,True] after data[1] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True,True,False] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is False of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is True of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is False and I will add it on the last element of data array so the data array has [True,False,True,False] after that I still add AND(a,b) in the last stack1 so the stack1 is [True,False] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal False of AND(Q,P) and the A variable is the last element of stack1 equal True of R. it's going on

the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is True and I will add it on the last element of data array so the data array has [True ,False,True,False,True] after that I still add OR(a,b) in the last stack1 so the stack1 is [True] with the OR operator of R and AND(Q,P).
-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [True ,False,True,False,True] and after that I will reset the data array.

The seventh row of table with i including True True False and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [True] and finishing the first loop so continue, the second loop with k = True so I add it in the last data so the data array has [True,True] and finishing the second loop. Continue the third loop with k = False so I add it in the last data so the data array has [True,True,False] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:
-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is False because the data array has [True,True,False] after data[2] is False then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False] of R and finish at the first loop is j.
-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is True because the data array has [True,True,False] after data[0] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False,True] of R,P and finish at the second loop is j.
-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order

2 and then we take a data of j from the data array via syntax data[1] so it is True because the data array has [True,True ,False] after data[1] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [False ,True,True] of R,P and Q and finish at the third loop is j.

-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is True of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is True of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is True and I will add it on the last element of data array so the data array has [True,True ,False,True] after that I still add AND(a,b) in the last stack1 so the stack1 is [False,True] with the first element of R and the second element is a and B like Q and P.

-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the A variable is the last element of stack1 equal True of AND(Q,P) and the A variable is the last element of stack1 equal False of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is True and I will add it on the last element of data array so the data array has [True,True ,False,True,True] after that I still add OR(a,b) in the last stack1 so the stack1 is [True] with the OR operator of R and AND(Q,P).

-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [True,True ,False,True,True] and after that I will reset the data array.

The last row of table with i including True True True and for k in i and then I will add each element of it in the last data array so the first loop of k is false so I add it in the last data via syntax data.append(k) so the data array has [True] and finishing the first loop so continue, the second loop with k = True so I add it in the last data so the data array has [True,True] and finishing the second loop. Continue the third loop with k = True so I add it in the last data so the data array has [True,True,True] and finishing the for loop of k in i because element i just has 3 element.

So I will use the for loop to browse the Postfix string wich j in Postfix:

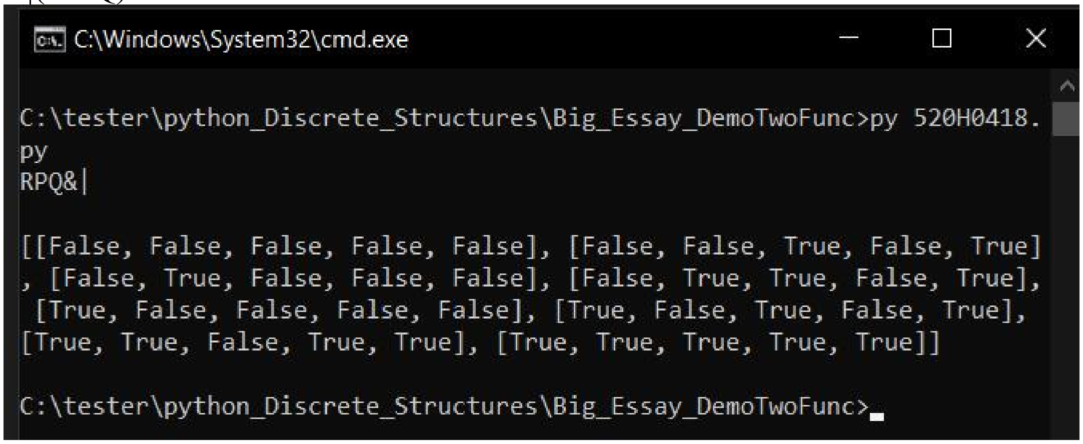
-The first loop of the Postfix with j = "R", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 2 because the stack has [P,Q,R] so j is order 3 and then we take a data of j from the data array via syntax data[2] so it is True because the data array has [True,True,True] after data[2] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True] of R and finish at the first loop is j.
-The second loop of the Postfix with j = "P", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 0 because the stack has [P,Q,R] so j is order 1 and then we take a data of j from the data array via syntax data[0] so it is True because the data array has [True,True,True] after data[0] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True ,True] of R,P and finish at the second loop is j.
-The third loop of the Postfix with j = "Q", it's going in the if condition at line 100 as it's alphabet characters so we will take a place of it and check where is the place of j in stack via syntax stack.index[j] so the order of j in stack which is 1 because the stack has [P,Q,R] so j is order 2 and then we take a data of j from the data array via syntax data[1] so it is True because the data array has [True,True ,True] after data[1] is True then I will add it in a last element of the stack1 array via syntax at line 101 so the stack 1 has [True ,True,True] of R,P and Q and finish at the third loop is j.
-The fourth loop of the Postfix with j = "&" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then I will create a variable which is a that contains an element of stack1 so I will take a last element of stack1 and the a variable is True of Q and at the same time, I also create a variable which is b that contains an element of stack1 so I will take a last element of stack1 and the b variable is True of P. it's going on the if elif condition at line 109 so it will calculate a logical operator which is "&" so AND(a,b) which is AND(Q,P), this result is True and I will add it on the last element of data array so the data array has [True,True ,True ,True] after that I still add AND(a,b) in the last stack1 so the stack1 is [True ,True] with the first element of R and the second element is a and B like Q and P.
-The fifth loop of the Postfix with j = "|" , it's going in the else condition at line 106 as it's different "~" and not a alphabet characters and then the

A variable is the last element of stack1 equal True of AND(Q,P) and the A variable is the last element of stack1 equal True of R. it's going on the if elif condition at line 112 so it will calculate a logical operator which is "|" so OR(a,b) which is OR(AND(Q,P),R), this result is True and I will add it on the last element of data array so the data array has [True,True ,True ,True,True] after that I still add OR(a,b) in the last stack1 so the stack1 is [True] with the OR operator of R and AND(Q,P).
-And finish the loop as j browsed all element of Postfix so I will add all element of the data array to a Truthtable so the Truthtable is [True,True ,True ,True,True] and after that I will reset the data array.

CHAPTER 4 – THE FOURTH

1.1 Testcase 1

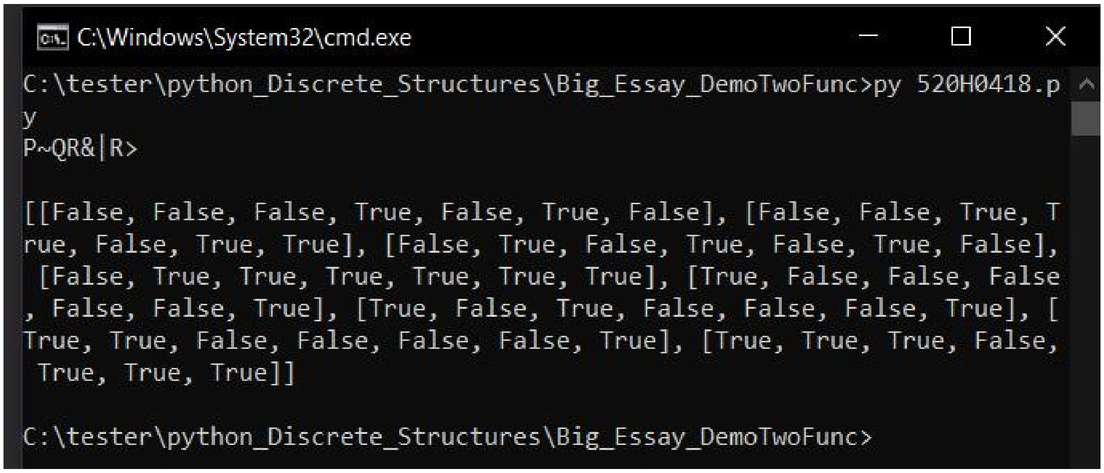
R|(P&Q)



```
C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>py 520H0418.
py
RPQ&|

[[False, False, False, False, False], [False, False, True, False, True]
, [False, True, False, False, False], [False, True, True, False, True],
 [True, False, False, False, False], [True, False, True, False, True],
[True, True, False, True, True], [True, True, True, True, True]]

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>
```

1.1 Testcase 2

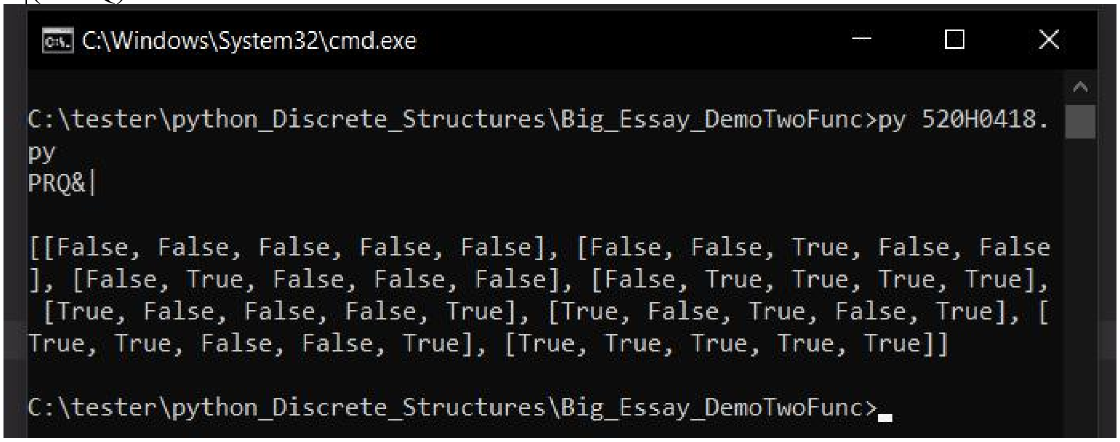
~P|(Q&R)>R



```
C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>py 520H0418.p
y
P~QR&|R>

[[False, False, False, True, False, True, False], [False, False, True, T
rue, False, True, True], [False, True, False, True, False, True, False],
 [False, True, True, True, True, True, True], [True, False, False, False
, False, False, True], [True, False, True, False, False, False, True], [
True, True, False, False, False, False, True], [True, True, True, False,
 True, True, True]]

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>
```

1.2 Testcase 3

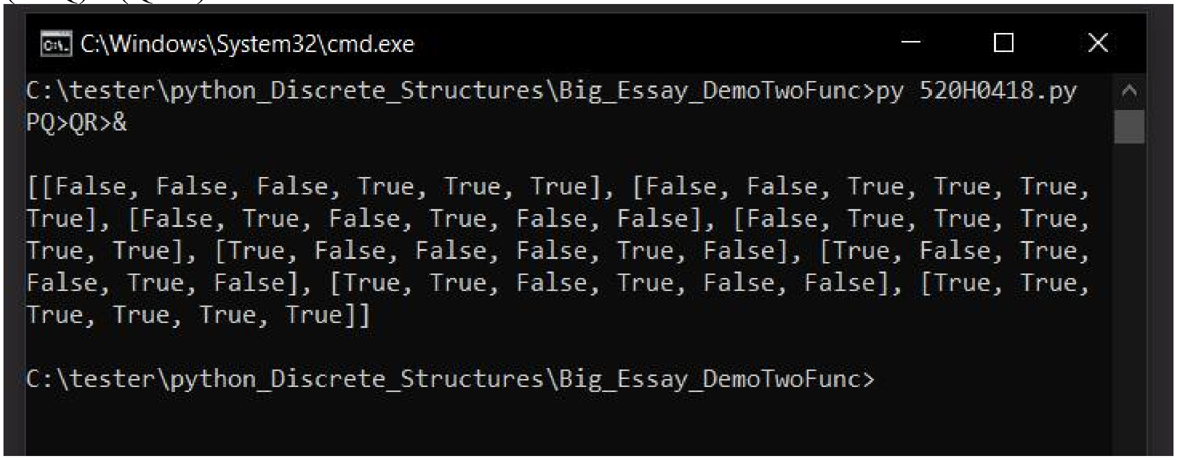
P|(R&Q)



```

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>py 520H0418.
py
PRQ&|

[[False, False, False, False, False], [False, False, True, False, False
], [False, True, False, False, False], [False, True, True, True, True],
 [True, False, False, False, True], [True, False, True, False, True], [
True, True, False, False, True], [True, True, True, True, True]]

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>
```

1.3 Testcase 4

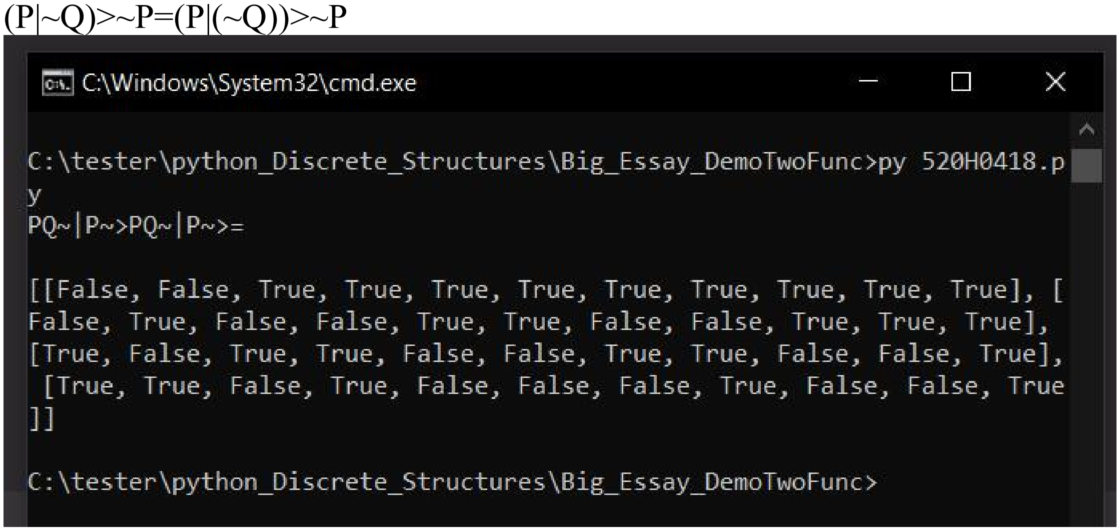
(P>Q)&(Q>R)



```
C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>py 520H0418.py
PQ>QR>&

[[False, False, False, True, True, True], [False, False, True, True, True,
True], [False, True, False, True, False, False], [False, True, True, True,
True, True], [True, False, False, False, True, False], [True, False, True,
False, True, False], [True, True, False, True, False, False], [True, True,
True, True, True, True]]

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>
```

1.4 Testcase 5

(P|~Q)>~P=(P|(~Q))>~P



```
C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>py 520H0418.p
y
PQ~|P~>PQ~|P~>=

[[False, False, True, True, True, True, True, True, True, True, True], [
False, True, False, False, True, True, False, False, True, True, True],
[True, False, True, True, False, False, True, True, False, False, True],
 [True, True, False, True, False, False, False, True, False, False, True
]]

C:\tester\python_Discrete_Structures\Big_Essay_DemoTwoFunc>
```

CHAPTER 5 – THE FIFTH

5.1 References

Reverse Polish: http://www-stone.ch.cam.ac.uk/documentation/rrf/rpn.html

Base logic: https://www.javatpoint.com/basic-logical-operations

THE END