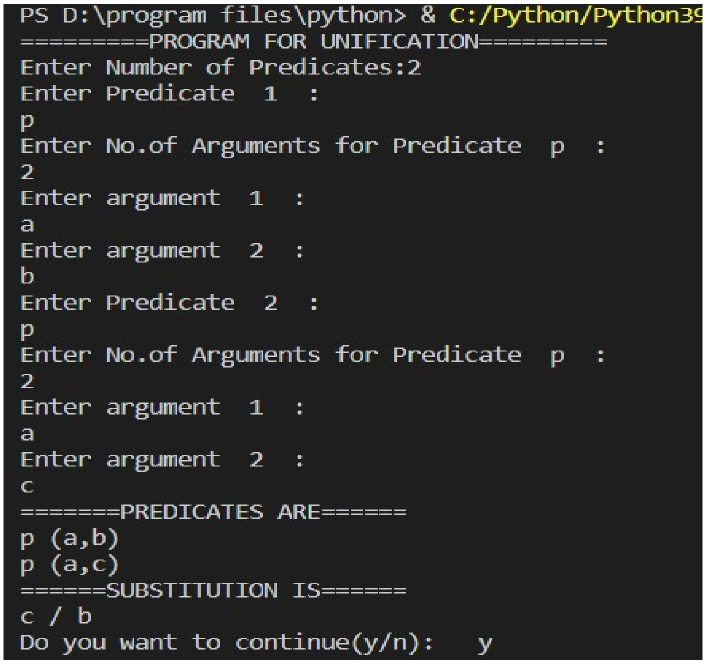
LAB8

CODE:

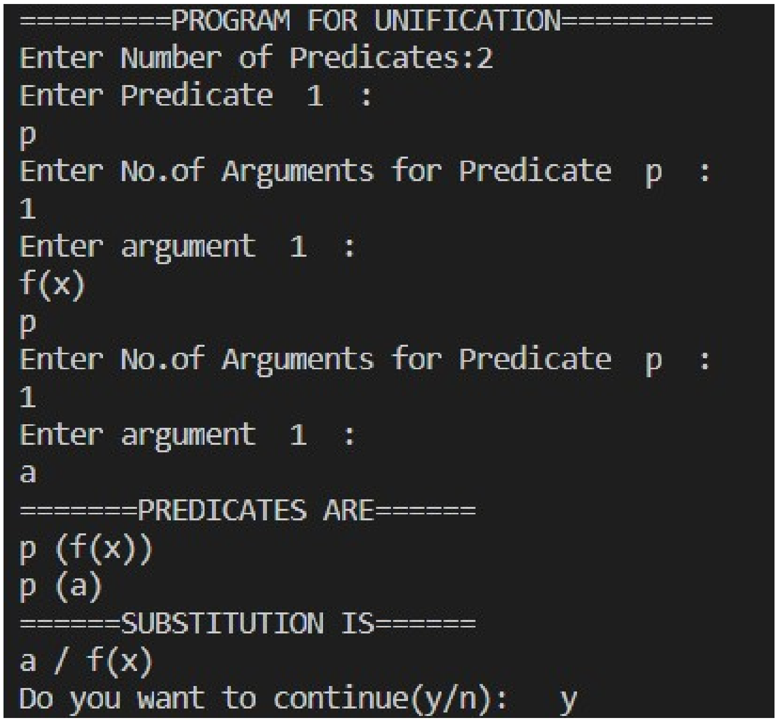
```
no_of_pred = 0
no_of_arg = [None for i in range(10)]

predicate = [None for i in range(10)]
argument = [[None for i in range(10)] for i in range(10)]

def main():
    global no_of_pred
    ch = 'y'
    while(ch == 'y'):
        print("=========PROGRAM FOR UNIFICATION=========")
        no_of_pred = int(input("Enter Number of Predicates:"))
        for i in range(no_of_pred):
            # nouse=input() #   //to accept "enter" as a character
            print("Enter Predicate ", (i+1), " :")
            predicate[i] = input()
            print("Enter No.of Arguments for Predicate ", predicate[i], " :")
            no_of_arg[i] = int(input())

            for j in range(no_of_arg[i]):
                print("Enter argument ", j+1, " :")
                argument[i][j] = input()

        display()
        chk_arg_pred()
        ch = input("Do you want to continue(y/n):   ")


def display():

    print("=======PREDICATES ARE======")
    for i in range(no_of_pred):
        print(predicate[i], "(", end="")
        for j in range(no_of_arg[i]):
            print(argument[i][j], end="")
            if(j != no_of_arg[i]-1):
                print(",", end="")
        print(")")


# /*==========UNIFY FUNCTION=========*/

def unify():
    flag = 0
    for i in range(no_of_pred-1):
        for j in range(no_of_arg[i]):
            if(argument[i][j] != argument[i+1][j]):
                if(flag == 0):
                    print("======SUBSTITUTION IS======")
                    print(argument[i+1][j], "/", argument[i][j])
                    flag += 1

        if(flag == 0):
            print("Arguments are Identical...")
            print("No need of Substitution")


def chk_arg_pred():
    pred_flag = 0
    arg_flag = 0

   # /*======Checking Prediactes========*/
    for i in range(no_of_pred-1):
        if(predicate[i] != predicate[i+1]):
            print("Predicates not same..")
            print("Unification cannot progress!")
            pred_flag = 1
            break

   # /*=====Chking No of Arguments====*/
```

```
    if(pred_flag != 1):
        ind = 0
        key = no_of_arg[ind]
        l = len(no_of_arg)
        for i in range(0, key-1):
            if i >= key:
                continue
            if ind != l-1:
                ind += 1
                key = no_of_arg[ind]
            if(no_of_arg[i] != no_of_arg[i+1]):

                print("Arguments Not Same..!")
                arg_flag = 1
                break

        if(arg_flag == 0 and pred_flag != 1):
            unify()

main()
```

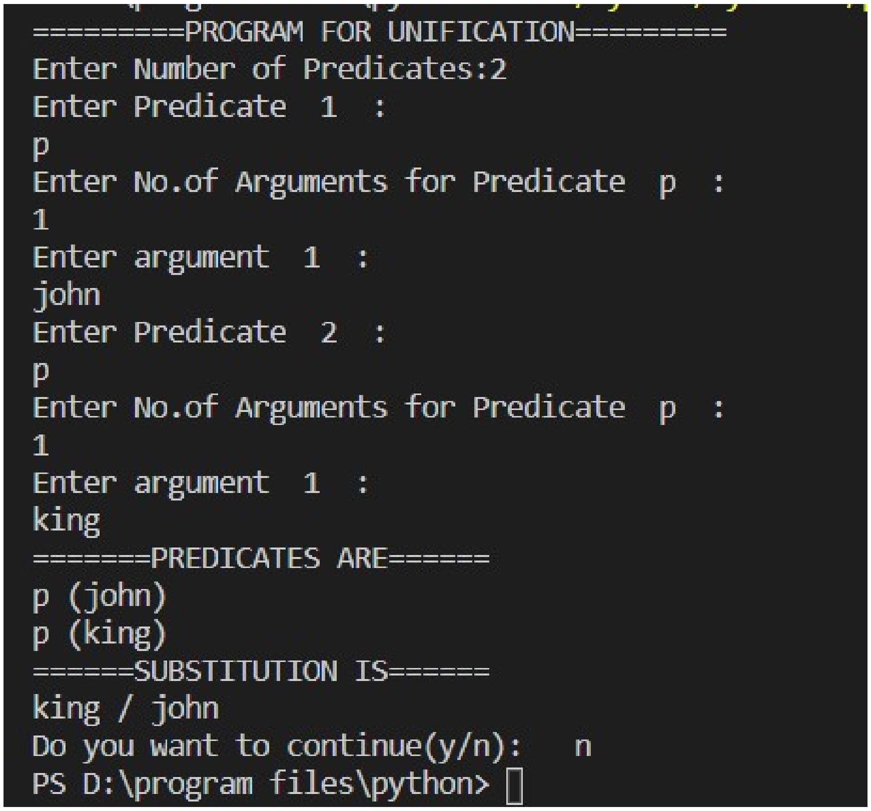
OUTPUTS



```
PS D:\program files\python> & C:/Python/Python3
=========PROGRAM FOR UNIFICATION=========
Enter Number of Predicates:2
Enter Predicate  1  :
p
Enter No.of Arguments for Predicate  p  :
2
Enter argument  1  :
a
Enter argument  2  :
b
Enter Predicate  2  :
p
Enter No.of Arguments for Predicate  p  :
2
Enter argument  1  :
a
Enter argument  2  :
c
=======PREDICATES ARE======
p (a,b)
p (a,c)
======SUBSTITUTION IS======
c / b
Do you want to continue(y/n):   y
```



```
=========PROGRAM FOR UNIFICATION=========
Enter Number of Predicates:2
Enter Predicate  1  :
p
Enter No.of Arguments for Predicate  p  :
1
Enter argument  1  :
f(x)
p
Enter No.of Arguments for Predicate  p  :
1
Enter argument  1  :
a
=======PREDICATES ARE======
p (f(x))
p (a)
======SUBSTITUTION IS======
a / f(x)
Do you want to continue(y/n):   y
```



```
=========PROGRAM FOR UNIFICATION=========
Enter Number of Predicates:2
Enter Predicate  1  :
p
Enter No.of Arguments for Predicate  p  :
1
Enter argument  1  :
john
Enter Predicate  2  :
p
Enter No.of Arguments for Predicate  p  :
1
Enter argument  1  :
king
=======PREDICATES ARE======
p (john)
p (king)
======SUBSTITUTION IS======
king / john
Do you want to continue(y/n):   n
PS D:\program files\python> 
```