Python Tracing

Tracing

Using added code or modifying code to find where the error is in the program.

There are numerous method, here are two:

- Print statements
- Commenting out

Print statements

Insert print statement at certain points in the program.

The print statements can put out a code , contents of the variable or both.

Code: print(‘a’)

Variable: print(rate) or print(‘rate:’,rate)

Both: print(‘a   rate:’,rate)

Example program

```
def valid(chk):
    if len(chk) == 0:
        return -1
    fchk = float(chk)
    if fchk < 0:
        return -1
    return fchk

def trace1():
    print('payroll calculator')
    hours = input('enter hours: ')
    rate = input('hourly rate; ')
    fhours = valid(hours)
    if fhours > 0:
        frate = valid(rate)
        if frate > 0:
            pay = fhours * frate
        else:
            print('invalid hours')
    else:
        print('invalid rate')
    print('Pay; ',pay)
trace1()
```

trace1

With trace

```
def valid(chk):
    if len(chk) == 0:
        return -1
    fchk = float(chk)
    if fchk < 0:
        return -1
    return fchk

def trace1():
    print('payroll calculator')
    hours = input('enter hours: ')
    rate = input('hourly rate; ')
    fhours = valid(hours)
    print('a  ')
    if fhours > 0:
        frate = valid(rate)
        print('b   ')
        if frate > 0:
            print('c')
            pay = fhours * frate
        else:
            print('d')
            print('invalid hours')
    else:
        print('e')
        print('invalid rate')
    print('f')
    print('Pay; ',pay)
trace1()
```

trace2

Comment out

Comment out line and run the program.

By commenting out, you will narrow down to the bad source statement in the program.

To comment out a statement use:

as first character

example

```
def spacer(a):
    print('\n'*a)
    return
def valid(chk):
    if len(chk) == 0:
        return -1
    fchk = float(chk)
    if fchk < 0:
        return -1
    return fchk
def trace3():
    print('payroll calculator')
    hours = input('enter hours: ')
    rate = input('hourly rate; ')
    fhours = valid(hours)
    print('--------------------')
    print('Number of hours worked: ', hours)
    print('Hourly rate of Pay: ',rate)
    print('--------------------')
    if fhours > 0:
        frate = valid(rate)
        if frate > 0:
            pay = fhours * frate
            print('Gross amout:  ',pay)
        else:
            print:('invalid hours')
    else:
        print('invalid rate')
    print('----------------------')
    return
def main():
    spacer(2)
    ans='y'
    while ans =='y':
        trace3()
        ans = input('Again y=yes, n=no -> ')
        spacer(2)
    return
main()
```

trace3

With comments

```
def spacer(a):
    #print('\n'*a)
    return
def valid(chk):
    if len(chk) == 0:
        return -1
    fchk = float(chk)
    if fchk < 0:
        return -1
    return fchk
def trace3():
    #print('payroll calculator')
    hours = input('enter hours: ')
    rate = input('hourly rate; ')
    fhours = valid(hours)
    #print('-------------------')
    #print('Number of hours worked: ', hours)
    #print('Hourly rate of Pay: ',rate)
    #print('-------------------')
    if fhours > 0:
        frate = valid(rate)
        if frate > 0:
            pay = fhours * frate
            #print('Gross amout:  ',pay)
        else:
            #print:('invalid hours')
    else:
        #print('invalid rate')
    #print('--------------------')
    return
def main():
    spacer(2)
    ans='y'
    while ans =='y':
        trace3()
        ans = input('Again y=yes, n=no -> ')
        spacer(2)
    return
main()
```

Trace4

Done