```python
#task 1
import math
result1=sum([.1, .1, .1, .1, .1, .1, .1, .1, .1, .1])
result2=math.fsum([.1, .1, .1, .1, .1, .1, .1, .1, .1, .1])
print(result1,"\n", result2)
'''
#0.9999999999999999
#1.0

the reason the results are different is because python stores the value of .1
as a different number, while in math.fsum trys to get us the closest exact
mathimatical sum representing number.
'''



#task 2
def task2() :
    for i in range (8) :
        print(i, end = ' ')
print(task2())

'''
# 0 1 2 3 4 5 6 7 None
the reason why the end says none is because the letter i
has already exceeded the range and can no longer increment in the range
'''



#task 3
#3a
def fun (x = 1, y = 2, z):
    z = x + y
    y += 1
    return z*y

fun(3,z=5)
'''
#when making function, without defining z, the program calls a syntax error.
#I believe any undefined variable has to be put in the beggining of the function
#    parameters and not following other defined variables
'''
#3b
def hoho (x, y = 2, z=1):
    z = x + z
    y += 1
    return z*y

print(hoho(5))
#no error- 18
print(hoho(6,z=3,y=1))
#no error- 18


#task 4
'''
def main () :
    z, y = 3, 4
    swap(x,y)
    print(x,y)
```

```
def swap(a, b):
    a, b = b, a


#x is not defined, replaced z with x
#main not called
#took out swap method and just returned value in reverse
#inorder to save the swap, you have to re define x and y to the new values
#revised below

'''

def main (x,y) :
    x,y= swap(x,y)
    print(x,y)

def swap(a, b):
    return(b,a)

main(1,2)
main(4,6)
"""
#2 1
#6 4
"""

#task 5
a, b = 0, 5             #global
def main() :
    global a, b         #global
    i = 10              #only in main
    b = g(i)            #b in main
    print(a+b+i)
def f(i) :
    n=0                 #n only in f()
    while n*n <= i:     #i is local
        n = n + 1
    return n-1
def g(a) :
    b=0                 #b only in g()
    for n in range(a):  #n only in g()
        i = f(n)        #i only in g()
        b = b+i         #b only in g()
        return b        #return in full loop, exits function before next loop

#potential answer '10'
main()
#real answer '10'
```