PS1_1

```
def Print_values(a,b,c):
    if (a>b):
        if (b>c):
            print (a,b,c)
        elif (b<c):
            if (a>c):
                print (a,c,b)
            elif (a<c):
                print (c,a,b)
    else:
        if (b>c):
            print ("The answer was not given")
        elif (b<c):
            print (c,b,a)
```

PS1_2

```
#2.1
#I got inspired by reading 'blog.csdn.net/weixin_39590868/article/details/113080569'
import numpy as np
M1=(np.random.randint(0,50,(5,10)))
M2=(np.random.randint(0,50,(10,5)))

#2.2
def Matrix_multip(M1,M2):
#I got inspired by reading
https://jingyan.baidu.com/article/22a299b51cf8d69e18376a57.html
    M3= np.zeros((5,5))
    c=[0,1,2,3,4]
    for a in c:
        l1=M1[a,:]
        for i in c:
            l2=M2[:,i]
            cl=l1*l2
            print(cl)
            num=0
#I got inspired by reading https://www.runoob.com/python3/python-sum-list.html
            for ele in cl:
                num = num + ele
            M3[a,i]=num
```

PS1_3

```
#I got inspired by reading https://www.cnblogs.com/findlisa/p/10179160.html
def Pascal_triangle(k):
    i=k
    lst1=[1]
    if i==1:
        print(lst1)
    elif i>1:
        print(lst1)
        num=1
        while num<k:
            lst2=lst1.append(0)
            cl=[lst1[i-1]+lst1[i] for i in range(len(lst1))]
            lst1=cl
            print(lst1)
            num=num+1
```

PS1_4

```
import numpy as np
num=int(input("enter an number:"))
if num==1:
    print(0)
else:
    #i got inspired by reading http://www.3qphp.com/python/pybase/2887.html
    M=[0 for i in range(num+1)]
    M[1]=0
    M[2]=1
    k=3
    while k<(num+1):
        if k%2!=0:
            M[k]=M[k-1]+1
        else:
            if M[k-1] > M[int(k/2)]:
                M[k] = M[int(k/2)]+1
            else:
                M[k] = M[k-1]+1
        k=k+1

    print(M[k-1]);
```

PS1_5

```
#I got inspired by reading
#https://blog.csdn.net/tao617/article/details/107547933?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_title~default-0.no_search_link&spm=1001.2101.3001.4242
#and discuss with Ding Chen and Zhan Yang

#5.1
def strings(n):
    if n ==1:
        return ['1']
    result = []
    for s in strings(n-1):
        result.append(s + str(n))
        result.append(s + "+" + str(n))
        result.append(s + '-' + str(n))
    return result

def find_expression(n,sum):
    solutions = strings(n)
    result = []
    for s in solutions:
        if eval(s) == sum:
            result.append(s + '=' + str(sum))
    return result

#5.2

total_solutions = find_expression(9,50) #9 为 1~9，50 为 random integer number

print("Total solutions are as follows:")
for solution in total_solutions:
    print(f"\n{solution}")

print(f"\nHere we have {len(total_solutions)} kinds of different solutions.")
```