

```
In [ ]: 1 # 1. Write a program that allows the user to enter any number of test scores. The user indicates they are done
2 by entering in a negative number. Print how many of the scores are A's (90 or above). Also print out the
3 average.
```

```
In [1]: 1 sl=[]
2 score=0
3 As=0
4 while score>=0:
5     score=int(input("Enter a score : "))
6     if score>=0:
7         if score>=90:
8             As+=1
9             sl.append(score)
10 print("Number of A's :",As)
11 print("avg of list: ",(sum(sl)/len(sl)))
```

```
Enter a score : 60
Enter a score : 60
Enter a score : 70
Enter a score : 70
Enter a score : 80
Enter a score : 80
Enter a score : 90
Enter a score : 90
Enter a score : 100
Enter a score : 100
Enter a score : -1
Number of A's : 4
avg of list: 80.0
```

```
In [ ]: 1 2. Write a program to determine how many zeroes 1000! ends with.
```

```
In [2]: 1 inp="1000!"
2 count=inp.count('0')
3 print(count)
```

```
3
```

```
In [ ]: 1 3. Use the following two lists and the format method to create a list of card names in the format card value
2 of suit name (for example, Two of Clubs).
```

```
In [3]: 1 suits = ["Hearts", "Diamonds", "Clubs", "Spades"]
2 values = ["One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King", "Ace"]
3 [print(values[j], " of ", suits[i]) for i in range(len(suits)) for j in range(len(values))]
```

```
One of Hearts
Two of Hearts
Three of Hearts
Four of Hearts
Five of Hearts
Six of Hearts
Seven of Hearts
Eight of Hearts
Nine of Hearts
Ten of Hearts
Jack of Hearts
Queen of Hearts
King of Hearts
Ace of Hearts
One of Diamonds
Two of Diamonds
Three of Diamonds
Four of Diamonds
Five of Diamonds
Six of Diamonds
```

```
In [ ]: 1 4. Write a program that creates the list [1,11,111,1111,...,111...1], where the entries have an ever
2 increasing number of ones, with the last entry having 100 ones.
```

```
In [6]: 1 num=1
        2 nl=[]
        3 for i in range(1,101):
        4     nl.append(i * "1")
        5 print(nl)
```

[illegible]

```
In [ ]: 1 5. Write a program that finds all pairs of six-digit palindromic numbers that are less than 20 apart. One such
        2 pair is 199991 and 200002
```

```
In [7]: 1 l1=[]
2 l2=[]
3 for x in range(100000,1000000):
4     z=len(str(x))//2
5     flag=0
6
7     for y in range(z):
8         if str(x)[y]!=str(x)[(-1-y)]:
9             flag=1
10
11         if flag==0:
12             l1.append(x)
13
14     for y in range(len(l1) - 1):
15         if abs(l1[y] - l1[y + 1]) < 20:
16             l2.append((l1[y], l1[y + 1]))
17
18 print(l2)
```

[(199991, 200002), (299992, 300003), (399993, 400004), (499994, 500005), (599995, 600006), (699996, 700007), (799997, 800008), (899998, 900009)]

```
In [ ]: 1 6. The number 99 has the property that if we multiply its digits together and then add the sum of its digits
2 to that, we get back to 99. That is,  $(9 \times 9) + (9 + 9) = 99$ . Write a program to find all of the numbers less than
3 10000 with this property. (There are only nine of them.)
```

```
In [8]: 1 for n in range(10001):
2       tot=0
3       prod=0
4       numlen=len(str(n))
5       for m in range(numlen):
6           tot+=int(str(n)[m])
7           if prod==0:
8               prod=int(str(n)[m])
9           else:
10              prod*=int(str(n)[m])
11       if (tot+prod)==n:
12          print(n)
```

```
0
19
29
39
49
59
69
79
89
99
```

```
In [ ]: 1 7. Write a program that finds all integer solutions to Pell's equation  $x^2-2y^2=1$ , where x and y are between
2 1 and 100.
```

```
In [9]: 1 ln=[]
2       ln = [(c, d) for c in range(1, 101) for d in range(1, 101) if (c**2)-(2*(d**2))==1]
3       print(ln)
```

```
[(3, 2), (17, 12), (99, 70)]
```

```
In [ ]: 1 8. Write a program to determine how many of the numbers between 1 and 10000 contain the digit 3.
```

```
In [13]: 1 count=0
2       for s in range(1,10001):
3           if '3' in str(s):
4               count+=1
5       print(count)
```

```
3439
```