assignment4.py > ...

```python
# Q 22-01

class Stack:  #implementation of STACK
    def __init__(self):...

    def push(self, element):...

    def pop(self):...

    def peek(self):...

    def is_empty(self):...

    def size(self):...

    def display(self):...

s=Stack()
s.push("Banana")
s.push("Apple")
s.push("Tomato")
s.pop()
s.push("Strawberry")
s.push("Grapes")
s.pop()
print(s.stack)

# OUTPUT :  ['Banana', 'Apple', 'Strawberry']

```

```
48
49  # Q 22-02
50
51  ss=Stack()
52  items=[10 * i for i in range(1,10)]
53  for item in items:
54      ss.push(item)
55      if(item//10)%2==0:
56          ss.pop()
57  print(ss.stack)
58
59  # OUTPUT :  [10, 30, 50, 70, 90]
60
61
```

```python

# Q 23-01

ss=Stack()
items=[10 * i for i in range(1,10)]
for item in items:
    ss.push(item)
    if(item//10)%2==0:
        ss.pop()
print(ss.stack)

# OUTPUT :  [10, 30, 50, 70, 90]


```

```
75
76  # Q 23-02
77  # QUEUE IMPLEMENTATION:
78  class Queue:
79      def __init__(self):...
81      def enqueue(self, element):...
84
85      def dequeue(self):...
93
94      def peek(self):...
101
102     def is_empty(self):...
104
105     def size(self):...
108
109     def display(self):...
111
112 q=Queue()
113 items=[10*i for i in range(1,11)]
114 for item in items:
115     q.enqueue(item)
116     if(item//10)%2==0:
117         q.dequeue()
118 print(q.queue)
119
120 # OUTPUT :: [60, 70, 80, 90, 100]
121
```

```
124
125 # Q 24-01
126
127 def find_two(nums):
128     x=y=0
129     for i in range(1, len(nums)):
130         if nums[x]< nums[i]:
131             x=i
132         elif nums[y]> nums[i]:
133             y=i
134     return x,y
135 nums=[11,37,45,26,59,28,17,53]
136 i,j=find_two(nums)
137 print(nums[i], nums[j])
138
139 # OUTPUT OF FUNCTION:: 4 , 0
140 #Output of Print::    59  11
141
```

```
141
142     # Q 24-02
143     # answer: The find_two() function performs 14 comparisons.
144
```

```python

# 25-01

from random import randint
maxi=int(input("Enter the max number:"))
number=int(input("Enter the guessing number:"))
c=0
l,h=1,maxi
while l<h:
    mid=(l+h)//2
    c+=1
    if mid==number:
        print(f"Your number is {number}")
        break
    elif mid> number:
        h=mid-1
    else:
        l=mid+1
print(f"Total {c} times are searched.")

# OUTPUT : Total 6 times are searched. c=6

# Q 25-02
# maximum = 100 and number =25
# ANSWER :: Total 2 times are searched.

```

```
171  # Q 26-01
172
173  table = HashTable()
174  book = "Alice in Wonderland"
175  key = sum(map(ord, book))
176  print(key, table.hash(key))
177
178  # KEY VALUE :  1763 <hash_value>
179
```

```python

# Q 26-02
class HashTable:...

table = HashTable()
books = ["The Old Man and the Sea",
         "The Little Prince",
         "Beauty and the Beast",
         "The Little Mermaid",
         "Alice in Wonderland"]

for book in books:
    table.add(book)

table.display()
# OUTPUT::
'''
Compartment 0: None
Compartment 1: None
Compartment 2: None
Compartment 3: Alice in Wonderland
Compartment 4: None
Compartment 5: None
Compartment 6: The Little Prince
Compartment 7: The Old Man and the Sea
Compartment 8: Beauty and the Beast
Compartment 9: The Little Mermaid

'''
```