10.17 Exercises
1. Yes

2. It starts from the end of the list .

3. [45 , -3 , 16 , 8]

4. (a) lst[0]
(b)lst[-1] / lst[3]
(c) 10
(d) 29
(e) -4
(f) 29
(g) 10
(h) It should be an integer or a slice to be legal

5. (a) 3
(b) 5
(c) 1
(d) 5
(e) 5
(f) 2
(g) 0
(h) 3
6. Yes

7. []

8. (a) [20,1,-34,40,-60,1,3]
(b) [20,1,-34]
(c) [-8,60,1,3]
(d) [-8,60,1,3]
(e) [40,-8]
(f) [20,1,-34]
(g) [-8,60,1,3]
(h) [20,1,-34,40,-8,60,1,3]
(i) [20,1,-34,40]
(j) [1,-34,40,-8]
(k) True
(l) False
(m) 8

9.

10. (a) [8,8,8,8]
(b) [2,7,2,7,2,7,2,7,2,7,2,7]
(c) [1,2,3,"a","b","c"]
(d) [1,2,1,2,1,2,4,2]
(e) [1,2,4,2,1,2,4,2,1,2,4,2]

11. (a) [3,5,7,9]
(b) [50,60,70,80,90]
(c) [12,15,18]
(d) [(0,0) , (0,1) , (0, 2), (0, 3), (1, 0), (1, 1), (1, 2), (1, 3), (2, 0), (2, 1), (2, 2), (2, 3)]
(e) [(0, 0), (0, 2), (1, 1), (1, 3), (2, 0), (2, 2)]

12. (a) [x**2 for x in range(1,6)]
(b) [x/4 for x in range(1,7)]

```
(c) [(x,y) for x in 'ab' for y in range(3)]
```

13. x in lst

14. Reverse iterator, so we can use it kind of instead of range .

15.
```
def sum_positive(a):
pos_sum = 0
for num in a:
pos_sum += num if num>0 else 0
return pos_sum
```

16.
```
def count_evens(a):
even_count = 0
for item in a:
even_count += 1 if item % 2 == 0 else 0;
return even_count
```

17.
```
def print_big_enough(lst, num):
for list_num in lst:
if list_num >= num:
print(list_num, end=' ');
```

18.
```
def next_number(lst):
lst_c = lst
num = 1
while True:
if len(lst_c)>0 and min(lst_c) == num:
lst_c.remove(num)
num += 1;
else:
break;
return num;
```

19.
```
def reverse(a):
return a[::-1]
```

20.
```
m = [[1]*9 for y in range(6)];
for row in m:
for item in row:
print(item, end=' ')
print()
print("----------------------")
m[2][4] = 0
print("----------------------")
for row in m:
for item in row:
print(item, end=' ')
print()
```

21.
```
1. lst = [1,2,3,4,5,6,7,8,9,10]
2. lst = [x for x in range(1,11)]
```

```
3. lst = list(range(1,11))
4. lst = [1,2] + list(range(3,6)) + [x for x in range(6,11)]
5. lst = []
for i in range(1,11):
lst += [i]

22.
def check_2d(list2d):
equal = False;
for i in range(len(list2d)):
if equal:
break;
row = list2d[i];
for j in range(len(list2d)):
if equal:
break;
column = [list2d[x][j] for x in range(len(list2d))];
if column == row :
equal = True;
break;
return equal;

23.
def check_winner(list2d):
for x in range(3):
if list2d[x][0] == list2d[x][1] == list2d[x][2] != '':
return list2d[x][0];
elif list2d[0][x] == list2d[1][x] == list2d[2][x] != '':
return list2d[0][x];
if list2d[0][0] == list2d[1][1] == list2d[2][2] != '':
return list2d[1][1];
elif list2d[0][2] == list2d[1][1] == list2d[2][0] != '':
return list2d[0][2];
return '';
```