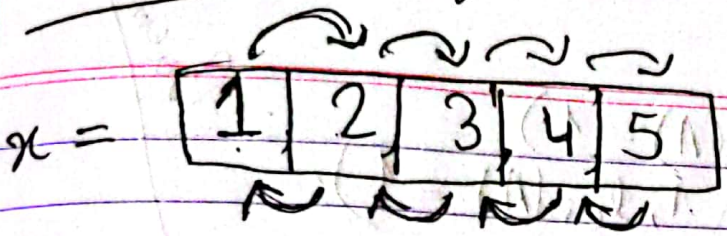
Lab assignment 1



2.

len = 5

shifts = 4

shifts = 3

k = 1

k = 2

k = 3

~~val = x[0]~~

new = [0] * k

```
i = 0
while (i <= k):
    new[i] = x[i]
    i += 1
```

~~k=4 k=1 m=2~~

```
j = 0
while (j <= len(x) - k - 1):
    x[j] = x[j+k]
    j += 1
```

~~m = 0~~
~~while (m <= len(new) - 1):~~
~~m = len(new) - 1~~
~~while (m >= 0)~~
~~x[len(x) - 1] = new[m]~~

```
m = 0
while (m < len(new)):
    x[len(x) - 1 - m] = new[len(new) - 1 - m]
    m += 1
```

Assignment - 1

Right shift

3.

shifting by k array = [10, 20, 30, 40, 50, 60]

i = len(array) - 1

```
while (i-k>=0):
    array[i] = array[i-k]
    i -= 1
for j in range(k)
    array[j] = 0
```

4.) Rotate ~~left~~ right

k = 3

[10, 20, 30, 40, 50, 60]

```
lost_array = [0]*k
i = len(array) - 1
for x in range(k)
while (len(array)-1
j=0
while (i>=k):
for j in range(k):
    lost_array[j] = array[i]
    j+=1
    i-=1
```

[60, 50, 60, 10, 20, 30]

[60, 50, 40]

```
n = len(source) - 1
while (n - k >= 0):
    array[n] = array[n-k]
    n -= 1
```

~~x =~~

```
for x in range(~~len(lost_arr)~~ k):
    array[x] = lost_array[k - x - 1]
```

~~Removal~~

size ind [10, 20, 30, 40, 50, 0, 0]
(5, 2)

[10, 20, 40, 50, 0, 0]

size = 5
ind = 2

```
str = ~~size - ind~~ ind + 1
while (str <= size):
    source[str-1] = source[str]
    str += 1
source[siz-1] = 0
```

6. size = 7

source = [10, 2, 30, 2, 50, 2, 2, 0]

i = 0

count = 0

while (i < size):

    if source[i] == 2:

        source[i] = 0

        count += 1

    i += 1

i = 7

count = 4

source = [10, 0, 30, 0, 50, 0, 0, 0]

for j in range(count):

    x = 0

    while (x < size):

        if source[x] == 0:

            source[x] = source[x+1]

source = [10, 30, 0, 50, 0, 0, 0, 0]

size = 7

i = 0

~~count = 0~~

[0, 2, 30, 2, 50, 2, 2, 0, 0]

1 2 3 4 5

while (i < size):

[10, 30, 2, 50, 2, 2, 2, 0, 0]

if source[i] == given:

~~count += 1~~

~~for j in~~

[10, 30, 50, 2, 2, 2, 2, 0, 0]

6

temp = i;

while (temp < size - 1):

source[temp] = source[temp + 1]

temp ~~+=~~ += 1

~~count = 4~~

count = 0

for j in range (size):

if source[j] == 2:

count += 1

count = 4

~~for~~ ~~while (size > str)~~ for j in range (count):

str = size - 1

source[str] = 0

str -= 1

7.

[1, 1, 1, 2, 1]

```
for i in range(len(source)-1):
    for j in range(i):
        total += source[j]
    for k in range(i+1, len(source)):
        total_2 += source[k]
    if total == total_2:
        print(True)
        break
    else:
        continue
elif i == len(source)-1:
    print(False)
    break
else:
    continue
```

n = 2, 3, 4,

21 = 4, 8

8.

$2^1, 2^2, 2^3$

n=2: {0 1 2 3 / 0, 1, 2, 1}

0, 1, 2

n=3: {0 1 2 3 4 5 6 8 / 0,0,1; 0,2,1; 3,2,1}

0, 1, 2, 3

n=4: {0 1 2 3 4 5 6 8 9 / 0,0,0,1; 0,0,2,1; 0,3,2,1; 4,3,2,1}

len_array = n*m

array = [0] * len_array

n = 3, 4

31 = 6, 9

for i in range (1, n+1):

for j in range (n-1+1)

8.

$n = 2$ [0, 1, 2, 1]

$n = 3$ [0, 0, 1, 0, 2, 1, 3, 2, 1]

$n = 4$ [0, 0, 0, 1, 0, 0, 2, 1, 0, 3, 2, 1, 4, 3, 2, 1]

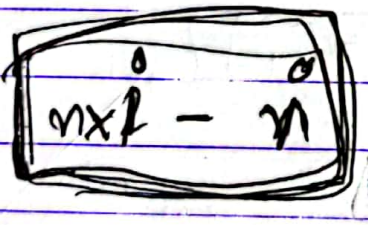
$n = 5$ [0, 0, 0, 0, 1, 0, 0, 0, 2, 1, 0, 0, 3, 2, 1, 0, 4, 3, 2, 1, 5, 4, 3, 2, 1]

$n = 3 / 4 / 5$

$2 = 4 / 6 / 8$

$n = 3 / 4 / 5$

$3 = 9 \quad 12$



Q10:

[1,2,2,3,4,4,4]

max = 0

for i in ~~rage~~ range(len(input)):

    if input[i] > max:

        max = input[i]

max = 4

new_array = [0]*(max+1)

0 1 2 3 4

[0,3,3,0,0]

for j in range(len(~~input~~ input)):

    ~~input[j]~~

    new_array[input[j]] += 1

~~max_2 = 0~~

~~ind = 0~~

~~for k in range(len(new_array)):~~

~~   if new_array[k] > max_2:~~

~~   max_2 = new_array[k]~~

~~   ind = k~~

~~print(ind)~~

for k in range(len(new_array)):

    val = new_array[k]

    for l in range(k+1, max+1):

        if new_array[l] == val:

            print(True)

            break

    else:

        continue

Q4

Q.

```
max = 0
count = 1

for i in range (len(input)):
    if input[i-1] = input[i]:
        count += 1
max_count = 0
count = 1
for i in range (1, len(input)):
    for j in range (i, len(input)):
        if input[j-1] = input[j]:
            max = input[j]
            count += 1
        else:
            break
    if max_count > count:
        max_count = count
    else:
        count = 1
```