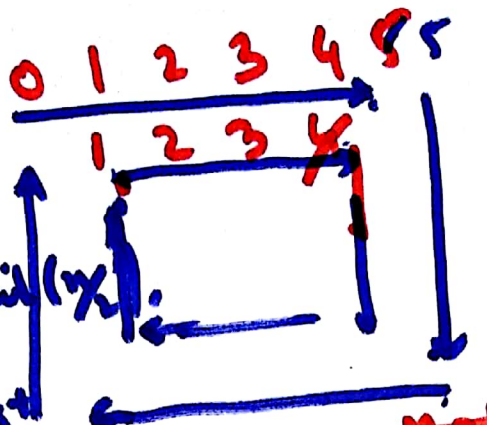
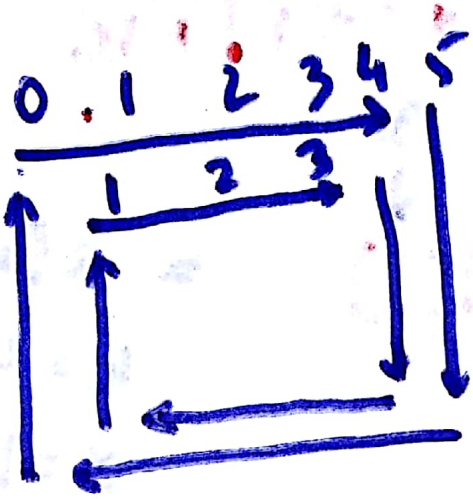



```
for(i=x, j=x; j<n-1; j++).
    printf("%d", a[i][j]);
for(i=x, i=x; i<n-1; i++).
    printf("%d", a[i][j]);
for(i=n-1; i>x; i--)
    printf("%d", a[i][j]);
for(i=x, i=n-1; i>x; i--).
    printf("%d", a[i][j]);
n=n-2;
n--;
}
```




```
ndash = n;
for( x=0; x<= ceil(ndash/2); x++, n--).
  {
  for(i=x j=x ; j<n-1; j++).
  for( ; i<n-1; i++).
   for( ; j>x; j--)
   for( ; i>x; i--).
  }
```

a[5][10] (10)

$x + (i * c + j) * k.$

2 → i, 3 → j

a[i][j]. = a[2][3]

k = 4

x = 15

$15 + (2 * 10 + 3) * 4.$

= ~~87~~ 107.

c = a b

c: [0 ...] = a: [[i,j] p,q] b: [[p,q]]

col ~~row~~ (a) = row (b).

c[p][q] = a[p][q] + b[p][q];

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} * b_{kj}$$

```
for (i
  for (i
    c[i][i] = 0;
    for (k=0; k<n; k++)
      c[i][i] += a[i][k] * b[k][i]
```

}

MALLOC A TWO-DIMENSIONAL ARRAY

```
int **a;
a = (int **) malloc (10 * sizeof (int *));
/* a[0], a[1], ... a[9] */
for (i=0; i<10; i++)
  a[i] = (int *) malloc (10 * sizeof (int));
```

```
int a[10][10];
```

```
free(a);
for (i=0; i<n; i++)
  free (a[i]);
free (a);
```