```
void displayValues(int[][MAX_COLS]) //read and display elements
{
	int i,j;
	for(i=0;i<MAX_ROWS;i++)
	{
		for(j=0;j<MAX_COLS;j++)
			printf("%4d ", t[i][j]);
		printf("\n");
	}
}
```

Add matrices

```
	int i,j;
	for (i=0;i<MAX_ROWS;i++)
	{
		for(j=0;j<MAX_COLS;j++)
			result[i][j]=a[i][j]+b[i][j];
	}
	displayValues(result);
```

Subtract matrices

```
	int i,j;
	for(i=0;i<MAX_ROWS;i++)
	{
		for(j=0;j<MAX_COLS;j++)
			result[i][j]=a[i][j]-b[i][j];
	}
	displayValues(result);
```

Exam's results are transferred into a function in a way of array composed of integers (integers represent grades). This array (flow), is consisted of grades 1, 2, 3, 4 and 5. Number of array's elements is nrGrades. Write your own function that returns most common grade.

```
int commonGrade(int *flow, int nrGrades)
{

	int grade[5] = {0}, commonGrade = 0, i;

	for(i = 0; i < nrGrades; i++)
		grade[flow[i] -1]++;

	for(i = 1; i < 5; i++)
	{
		if(grade[i] > grade[commonGrade])
			commonGrade = i;
	}
```

```
	return commonGrade + 1;
}
```

Example:

Write your own function that shifts array p (given, integer type) for a number of places. Array p is consisted of N number of elements, and will be shifted left for negative value, or shifted right for positive value (adjustment < N). Empty places are filled with zeroes.

Example of shifting:

shift = 2 and array[10 40 50 60 12]:
Before shifting: [10 40 50 60 12]
Shifted array: [0 0 10 40 50]

```
void shiftArray(int *p, int N, int shift)
{
	int i;

	/* if shifting right */
	if (shift > 0 && shift < N)
	{
		for (i = N - 1 - shift; i >= 0; i --)
			p[i + shift] = p[i];
		for (i = shift - 1; i >= 0; i --)
			p[i] = 0;
	}
	/* if shifting left */
	else if (shift < 0 && -shift < N)
	{
		shift = -shift;
		for (i = 0; i <= N - 1 - shift; i++)
			p[i] = p[i + shift];
		for (i = N - shift; i < N; i++)
			p[i] = 0;
	}
}
```