```
public class FæraUmEinn {
    public static void main(String[] args) {
        int N = args.length;
        String[] b = new String[N];
            for (int i = 0; i < N-1; i++) {
                b[i] = args[i+1];   }
        b[N-1] = args[0];
        for (int i = 0; i < N; i++) {
            System.out.println(args[i]); }
        System.out.println("******");
        for (int i = 0; i < N; i++) {
             System.out.println(b[i]); }}}
```

RAGGED ARRAY

```
for (int i = 0; i < a.length; i++)
{
   for (int j = 0; j < a[i].length; j++)
      System.out.print(a[i][j] + " ");
   System.out.println();
}
```

expression	*expression type*	*expression value*
"1234" + 99	String	"123499"
Integer.parseInt("123")	int	123
(int) 2.71828	int	2
Math.round(2.71828)	long	3
(int) Math.round(2.71828)	int	3
(int) Math.round(3.14159)	int	3
11 * 0.3	double	3.3
(int) 11 * 0.3	double	3.3
11 * (int) 0.3	int	0
(int) (11 * 0.3)	int	3

```
public class BoolArray {
    public static void main(String[] args) {
        int M = Integer.parseInt(args[0]);
        int N = Integer.parseInt(args[1]);
        boolean[][] t = new boolean[M][N];
        for (int i = 0; i < M; i++) {
            for (int j = 0; j < N; j++) {
                if (Math.random() < 0.5) t[i][j] = true;
                else t[i][j] = false;     }
             }
        for (int i = 0; i < M; i++) {
            for (int j = 0; j < N; j++) {
                if (t[i][j] == true) System.out.print("* ");
                else System.out.print("_ "); }
             System.out.println(); }}}
```

```
public class SlembiDagurSwitch {
      public static void main(String[] args) {
            int sdag = (int)(Math.random()*5.0);
            switch (sdag) {
                  case 0: System.out.println("Mánudagur");
                        break;
                  case 1: System.out.println("Þriðjudagur");
                        break;
                  case 2: System.out.println("Miðvikudagur");
                        break;
                  case 3: System.out.println("Fimmtudagur");
                        break;
                  case 4: System.out.println("Föstudagur");}}}
```

```
int i, j;
for (i=0, j=0; i<10; i++)
j += j++;
```

Lokagildi j er 0, því í setningunni "j += j++", þá er upphafsgildi j notað til að hækka j (sem er 0), síðan er j hækkað um 1, en vinstri hlið setningarinnar fær gildið á eftir, svo það fær alltaf gildið 0.

hvað gefur $1/1^2+/2^2+...+1/N^2$?

```
a) for (int i = 0; i <= N; i++) sum += 1 / (i*i);
b) for (int i = 0; i <= N; i++) sum += 1.0 / i*i;
c) for (int i = 0; i <= N; i++) sum += 1.0/ (i*i);
d) for (int i = 0; i <= N; i++) sum += 1 / (1.0*i*i);
```

a) Þar sem i er skilgreind sem int-breyta, þá verður yfirflæði (overflow) þegar við reiknum há gildi á "(i*i)". Stærsta jákvæða gildi sem kemst í int-breytu er 2,147,483,647. Það fæst þegar i er komið í rúmlega 46,340. Við fáum reyndar deilingu með núlli, því yfirflæðið veldur því að eitt margfeldið verður 0.
b) Þessi lykkja gefur rangt svar (sum er 1000000.0), því það vantar sviga fyrir neðan strik í deilingunni.
c) Hér er aftur yfirflæði á i*i, en vegna þess að deilingin er gerð í kommutölum, þá fáum við niðurstöðuna $+\infty$, því það er útkoman í kommutölum þegar deilt er með núlli.

public class Math

`double abs(double a)`	*absolute value of a*
`double max(double a, double b)`	*maximum of a and b*
`double min(double a, double b)`	*minimum of a and b*

Note 1: `abs()`, `max()`, *and* `min()` *are defined also for* `int`, `long`, *and* `float`.

`double sin(double theta)`	*sine function*
`double cos(double theta)`	*cosine function*
`double tan(double theta)`	*tangent function*

Note 2: Angles are expressed in radians. Use `toDegrees()` *and* `toRadians()` *to convert.*
Note 3: Use `asin()`, `acos()`, *and* `atan()` *for inverse functions.*

`double exp(double a)`	*exponential* (e^a)
`double log(double a)`	*natural log* ($\log_e a$, *or* $\ln a$)
`double pow(double a, double b)`	*raise a to the bth power* (a^b)
`long round(double a)`	*round to the nearest integer*
`double random()`	*random number in* [0, 1)
`double sqrt(double a)`	*square root of a*
`double E`	*value of e (constant)*
`double PI`	*value of π (constant)*

create an array with random values	`double[] a = new double[N];`
	`for (int i = 0; i < N; i++)`
	`  a[i] = Math.random();`

print the array values, one per line	`for (int i = 0; i < N; i++)` `  System.out.println(a[i]);`

find the maximum of the array values	`double max = Double.NEGATIVE_INFINITY;`
	`for (int i = 0; i < N; i++)`
	`  if (a[i] > max) max = a[i];`

compute the average of the array values	`double sum = 0.0;`
	`for (int i = 0; i < N; i++)`
	`  sum += a[i];`
	`double average = sum / N;`

copy to another array	`double[] b = new double[N];`
	`for (int i = 0; i < N; i++)`
	`  b[i] = a[i];`

reverse the elements within an array	`for (int i = 0; i < N/2; i++)`
	`{`
	`  double temp = b[i];`
	`  b[i] = b[N-1-i];`
	`  b[N-i-1] = temp;`
	`}`

public class StdIn

`boolean isEmpty()`	`true` *if no more values,* `false` *otherwise*
`int readInt()`	*read a value of type* `int`
`double readDouble()`	*read a value of type* `double`
`long readLong()`	*read a value of type* `long`
`boolean readBoolean()`	*read a value of type* `boolean`
`char readChar()`	*read a value of type* `char`
`String readString()`	*read a value of type* `String`
`String readLine()`	*read the rest of the line*
`String readAll()`	*read the rest of the text*

API for our library of static methods for standard input

type	*code*	*typical literal*	*sample format strings*	*converted string values for output*
`int`	`d`	`512`	`"%14d"` `"%-14d"`	`"  512"` `"512  "`
`double`	`f`	`1595.1680010754388`	`"%14.2f"` `"%.7f"`	`"  1595.17"` `"1595.1680011"`
	`e`		`"%14.4e"`	`"  1.5952e+03"`

`String`	`s`	`"Hello, World"`	`"%14s"`
			`"%-14s"`
			`"%-14.5s"`
				`"  Hello, World"`
				`"Hello, World  "`
				`"Hello  "`

`int`	`Integer.parseInt(String s)`	*convert* `s` *to an* `int` *value*
`double`	`Double.parseDouble(String s)`	*convert* `s` *to a* `double` *value*
`long`	`Long.parseLong(String s)`	*convert* `s` *to a* `long` *value*

```
public class MaxMin {
    public static void main(String[] args) {

        // first value read initialized min and max
        int max = StdIn.readInt();
        int min = max;

        // read in the data, keep track of min and max
        while (!StdIn.isEmpty()) {
            int value = StdIn.readInt();
            if (value > max) max = value;
            if (value < min) min = value;
        }

        // output
        StdOut.println("maximum  = " + max + ", minimum = " + min);
    }
}
```