WEEK 3

YOU'RE STILL HERE?

TODAYS PLAN

Administrative Stuff

- Structural Changes
- Submission
- Homework Correction
- Note to UTF8

Tutor Tasks

- Syntax Trees
- Flow Charts
- MinMax
- Pascals Triangle

ADMINISTRATIVE

ADMINISTRATIVE – STRUCTURAL CHANGES

Time	Beginners	Experienced
15 Mins	Admin + Homework Discussion Time	
30 Mins	Task 1: General non-programming skills	
30 Mins	Task 2: General non-programming skills	
15 Mins	General Explanation to Tasks 3 and 4 (Programming Tasks)	
15 Mins	Theory Explanation Task 3	Own Programming Time Questions can be asked during Programming Time
30 Mins	Programming Time Task 3	
15 Mins	Theory Explanation Task 4	
30 Mins	Programming Time Task 4	

ADMIN – SUBMISSIONS

ADMIN – SUBMISSIONS

- 3 Instances in Moodle has been disabled
  - See your email
- Submit in the Main Instance

ADMIN – HOMEWORK CORRECTION

- Thursday Afternoon I shall begin correcting
  - Model Solutions uploaded
  - Most questions have been asked in the tutor group

ADMIN – LAST WEEK'S HOMEWORK

- Any Questions?

ADMIN – UTF8

ADMIN – UTF8

- Linux → To be absolutely sure use “file –bi <filename>”
- Mac → Probably the same as Linux
- Windows → Atom Text Editor

TUTOR TASKS

01

Syntax Trees

02

Flow Charts

03

MinMax

04

Pascals Triangle

SYNTAX TREES

SYNTAX TREES

- A lot of effort, but easy exam points

SYNTAX TREES

- Grammar on Moodle

SYNTAX – TREES DEMO

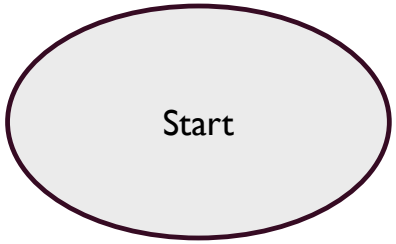
- Sum Function
- Absolute Value Function

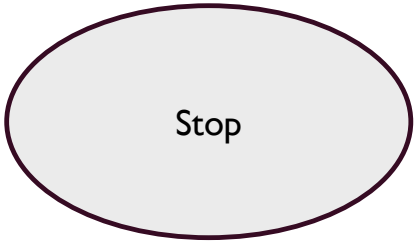
SYNTAX TREES – YOUR TURN

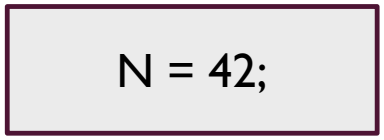
```
int prod, x, n;
x = read();
if (0 < x) {
  prod = 1;
  n = 0;
  while (prod <= x) {
    n = n + 1;
    prod = prod * (-n);
  }
  write(prod);
} else {
  write(n);
}
```

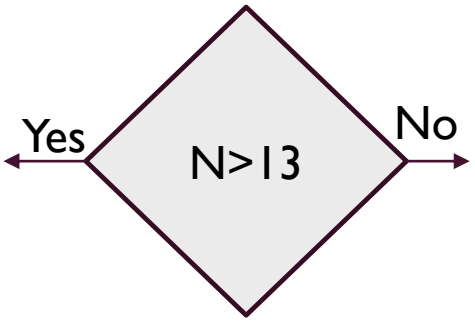
FLOW CHARTS

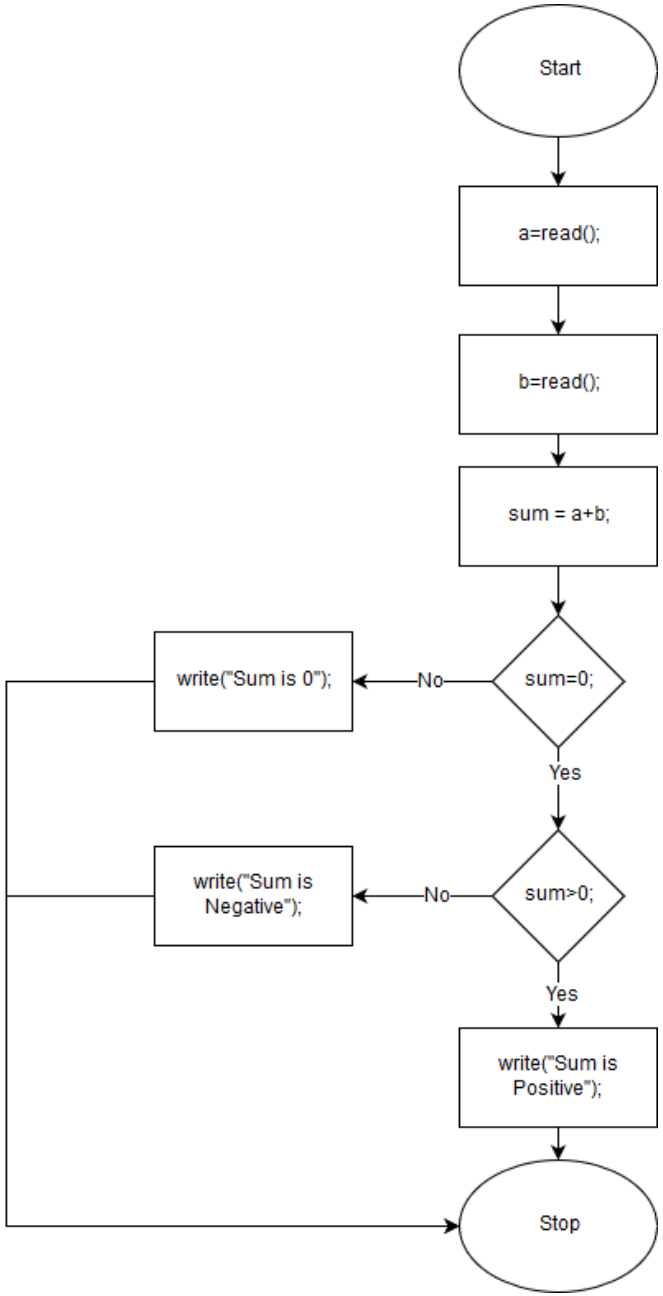
FLOW CHARTS – SYMBOLS











FLOW CHART

EXAMPLE

FLOW CHART – SITE

- www.draw.io

```
int prod, x, n;
x = read();
if (0 < x) {
  prod = 1;
  n = 0;
  while (prod <= x) {
    n = n + 1;
    prod = prod * (-n);
  }
  write(prod);
} else {
  write(n);
}
```

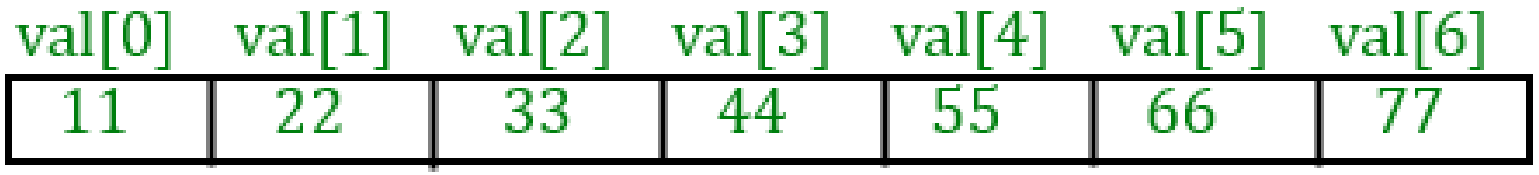
FLOW CHART

CODE

MIN-MAX

MINMAX – ARRAYS

- Essentially an Array is a list of Objects of a certain type



MINMAX – DECLARING AN ARRAY

- int[] arr = new int[5] → {0, 0, 0, 0, 0}
- int[] arr = {1, 2, 3, 4, 5} → {1, 2, 3, 4, 5}

MINMAX – USEFULL ARRAY FUNCTIONS

Say int[] n = new int[arrlength];

- n.length() → returns arrlength

MINMAX – ARRAY DEMO

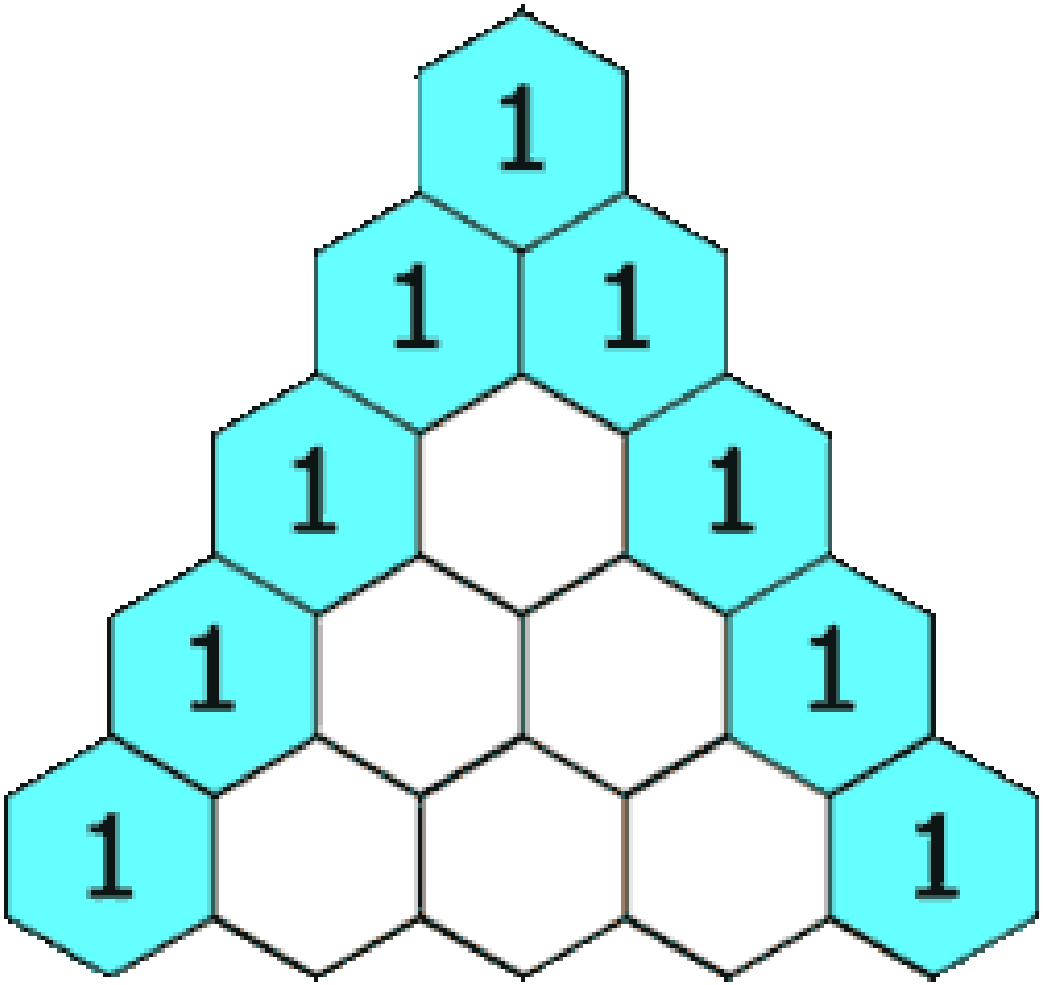
MINMAX – YOUR TASK

- Find the smallest and largest elements of an array

MINMAX – YOUR TASK

1. Ask user for array length
2. Ask user for all values
3. In **one** pass through the smallest & largest numbers should be found
4. Output the largest and smallest numbers

PASCAL'S TRIANGLE



PASCALS TRIANGLE

PASCALS TRIANGLE

- Binomial Coefficient

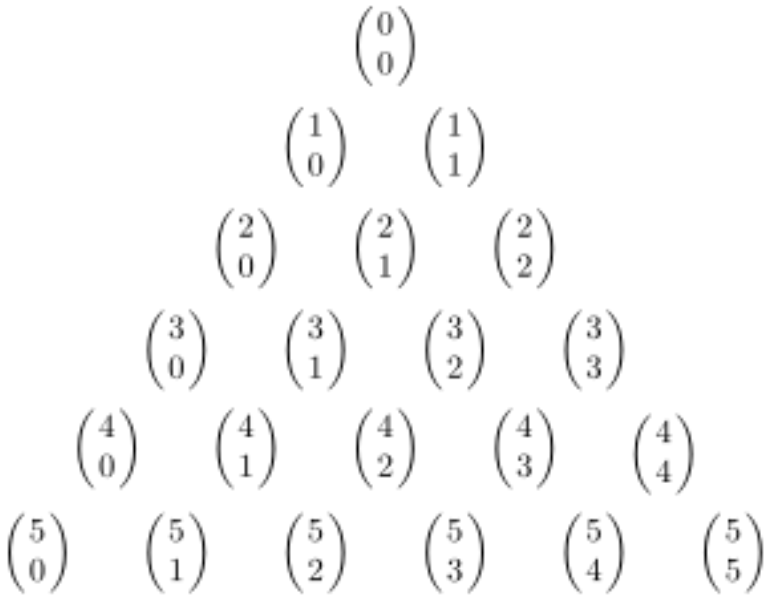
$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$\begin{array}{ccccccccccc}
 & & & & & \binom{0}{0} & & & & & \\
 & & & & \binom{1}{0} & & \binom{1}{1} & & & & \\
 & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & \\
 & & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & \\
 & \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & \\
\binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5}
\end{array}$$

PASCALS TRIANGLE

- Binomial Coefficient
  $n \geq k \geq 0$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

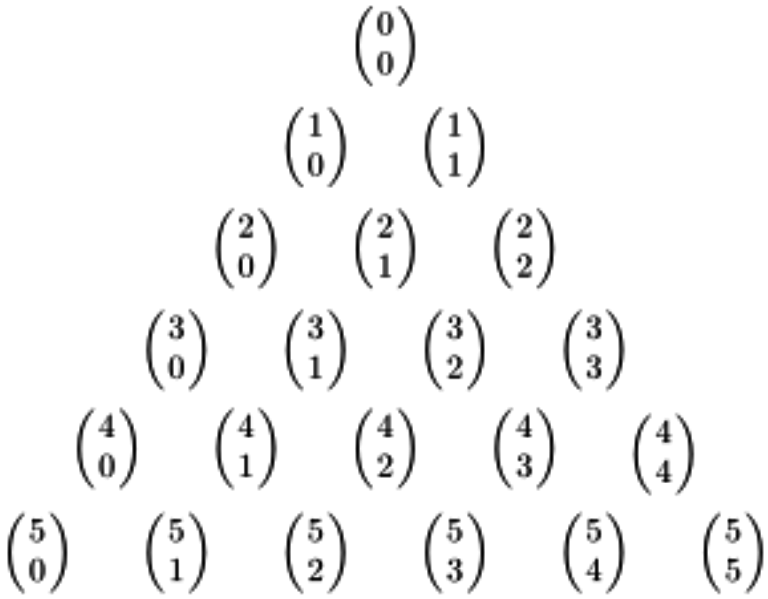


How many of you still remember how to calculate factorials?
What is 5!

PASCALS TRIANGLE

- Binomial Coefficient
  $n \geq k \geq 0$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

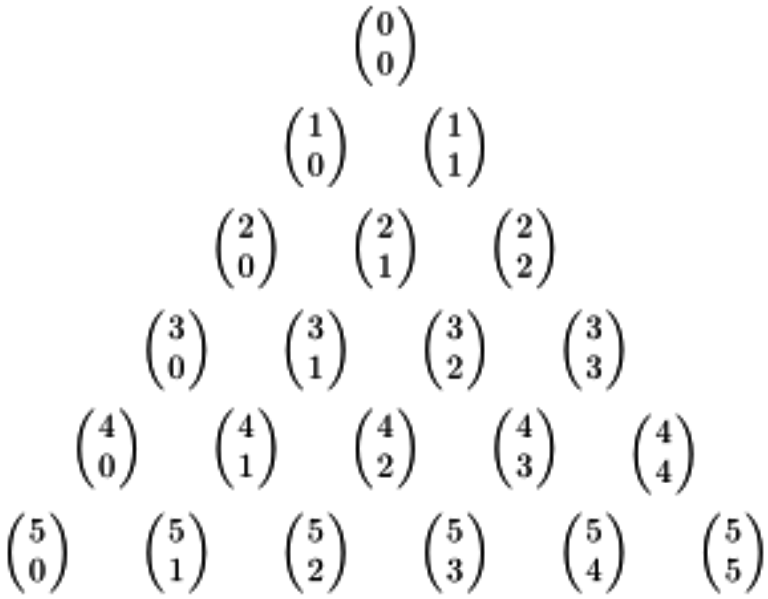


How many of you still remember how to calculate factorials?
5! = 5*4*3*2*1=120

PASCALS TRIANGLE

- Binomial Coefficient
  $n \geq k \geq 0$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$



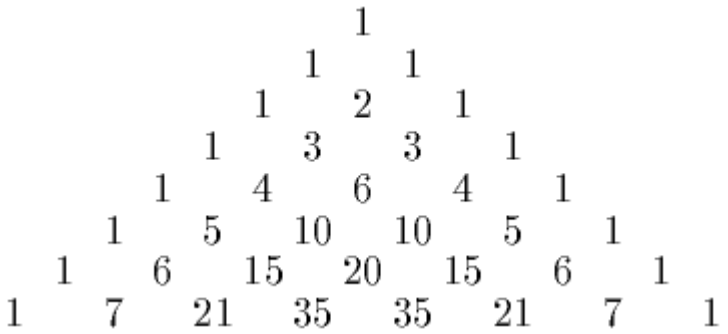
How many of you still remember how to calculate factorials?
5! = 5*4*3*2*1=120

PASCALS TRIANGLE

Pascalsches Dreieck mit 5 Zeilen:

Zeilenindex n					
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

1. Get a row value for n from the user
2. Generate pascals triangle rows from 0 to n-1
   Each row shall be put into an array
3. Output the array in a sensible fashion



PASCALS TRIANGLE

- Prompt the user a new input if the number is smaller than 1

PASCALS TRIANGLE – TIPS

- First and Last element of an array is always 1