



# Engineering program development 2

Edited by Péter Vass

# Designing and expressing algorithms

## Task I:

Let us summarize the positive integers from 1 to 10 by means of a pre-test loop and display the result.

$$s = 1+2+3+4+\dots+10$$

$$s = 1$$

$$s = s+2$$

$$s = s+3$$

....

$$s = s+10$$

## Analysis of the problem:

- the limits of the range are fixed by two constants,
- we should introduce a variable (s) for storing the intermediate result of each addition,
- we have to use a pre-test loop for providing the repetition of addition
- inside the loop body, we must add the next member of the series to the intermediate result of the previous step,
- we should introduce another variable for storing the next member of the series (x),

# Designing and expressing algorithms

## Analysis of the problem:

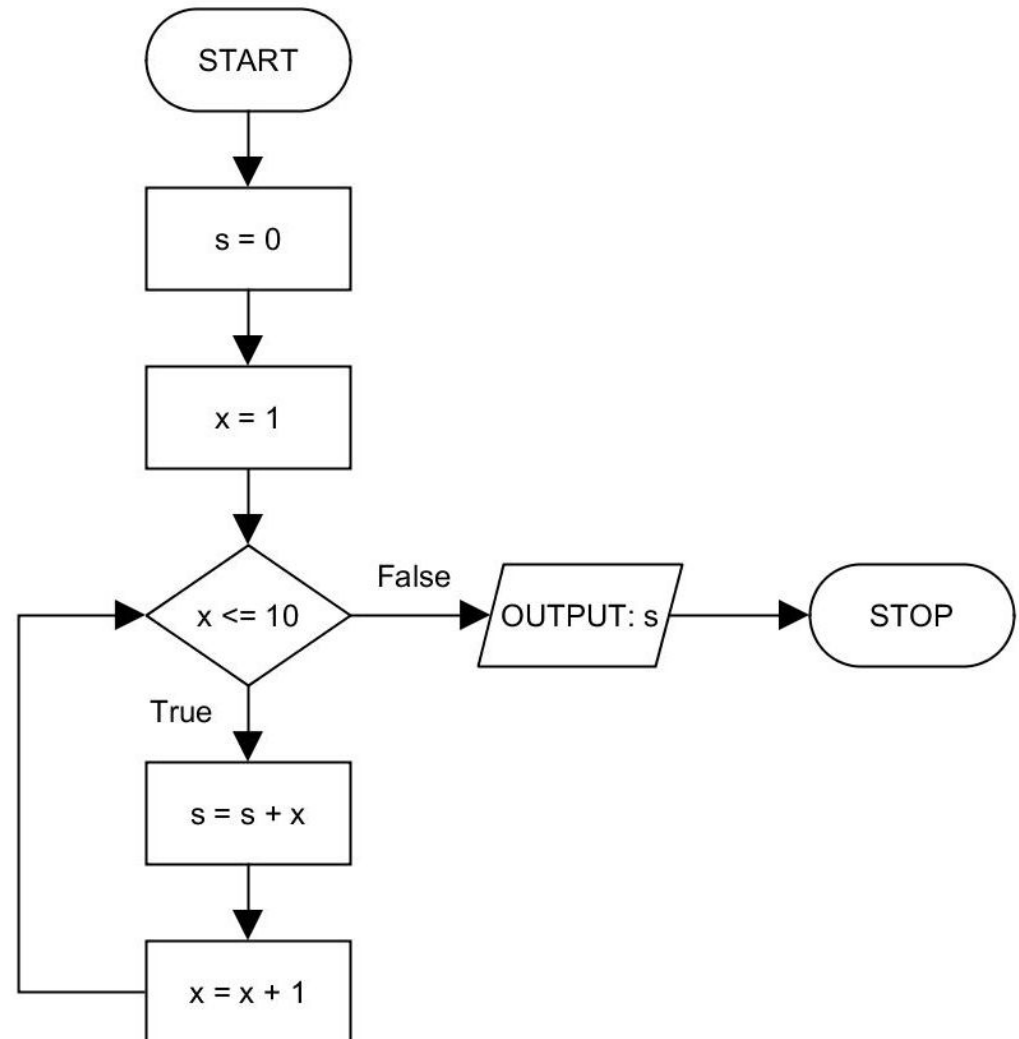
- variable  $x$  is actually a loop variable whose value must be incremented by one inside the loop body after the addition,
- the condition in the loop control statement is applied to the actual value of  $x$  which may not be greater than 10 ( $x \leq 10$ ).

Now we can make the pseudocode and flowchart of the algorithm.

# Designing and expressing algorithms

The pseudocode and flowchart of the algorithm

```
s = 0
x = 1
while x <= 10 do
    s = s + x
    x = x + 1
write s
```



# Creating flowcharts by yED Graph Editor

- two modes can be selected in the editor window:  
edit mode (to create graphs and modify their elements),  
navigation mode (to move the viewpoint of the window)  
View → Mode or from the toolbar,
- if the navigation mode is active, the graphical objects cannot be selected and modified, but the viewpoint can be moved by left or right mouse drag,
- a grid can be switched on or off in the editor window: View → Grid or from the tool bar,
- adjusting the type and spacing of grid: File → Preferences → Grid,
- pre-defined elements of flowchart: Palette window → Flowchart,
- creating the new element of a flowchart: left mouse dragging the representation of the element from the Palette window.

# Creating flowcharts by yED Graph Editor

- selecting an existing element in the editor window: left mouse click on the element,
- deleting a selected element: pressing DEL on the keyboard or Edit → Delete or from the tool bar,
- modifying the properties of a selected element: inside the Properties View window or right mouse click on the selected element → Properties item from the pop-up menu
- the most frequently modified properties:
  - Text (the text displayed in the element),
  - Fill Color, Fill Color 2 (a shaded filling can be created by selecting two different colours for them),
  - Line Color (the colour of the border line),
  - X and Y (the coordinates of upper-left corner of the element),
  - Width and Height (the width and height of the element),

# Creating flowcharts by yED Graph Editor

- the position and size of a selected element may be modified by means of the mouse (left mouse dragging of the entire element or one of the eight knobs around it)
- a selected element in the editor window may be duplicate by means of Ctrl+C & Ctrl+V technique,
- linking two elements by a line:  
do not select either of the elements,  
draw a line from the source element to the target by left mouse dragging (start over the source and finish over the target),
- the position of line may be changed afterwards
- the properties of a selected line may be modified in the Properties View window or the pop-up menu activated by right mouse click

# Creating flowcharts by yED Graph Editor

- the most frequently modified properties:  
Line Color, Line Type, Source Arrow (the type of arrow pointing the source element), Target Arrow (the type of arrow pointing the target element).
- a line with break-points may also be created:  
the line must be started over a source element by left mouse drag,  
the mouse button must be released to create the first break-point,  
then each subsequent left mouse click creates a further break-point,
- the segmented line must be finished over a target element by a left mouse click.
- to break the line rectangularly, the function called Orthogonal Edges must be switched on: View → Orthogonal Edges, or from the tool bar,



# Creating flowcharts by yED Graph Editor

- a label (with text) may also be attached to a line by means of adjusting the Text property of a selected line,
- the position and other properties of a selected label may be modified,
- setting by which the position of a label may be modified freely: Properties View window → Placement property → Modell = Free and Position: = Anywhere,
- saving a graph: File → Save or from the tool bar
- other useful functions:  
zooming in and out (View or tool bar)
- Structure View window lists all the elements building up a graph (even an overlapped element may be selected in that window).

# Creating flowcharts by yED Graph Editor

- Neighbourhood window shows the near environment of a selected element,
- Overview window provides an overall view about the graph,
- several elements may be selected collectively (Shift + left mouse click or enclosing the elements by left mouse drag),
- the selected elements may be grouped: Grouping → Group
- the grouped elements may collectively be moved,
- decomposing a selected group: Grouping → Ungroup,
- exporting a graph in different file formats: File → Export → selection of folder and file format → specifying the file name → adjusting the size of image or the value of Scaling factor

# Designing and expressing algorithms

## Task 2:

Let us multiply the positive even numbers from 1 to 10 by means of a post-test loop, and display the result.

## Task 3:

Let us calculate the sum below and display the result:

$$s = \sum_{i=1}^n (-1)^i \cdot \frac{i-1}{i+1}$$

## Task 4:

Let us calculate the product below, and display the result:

$$q = \prod_{i=1}^n \frac{i}{i+1}$$

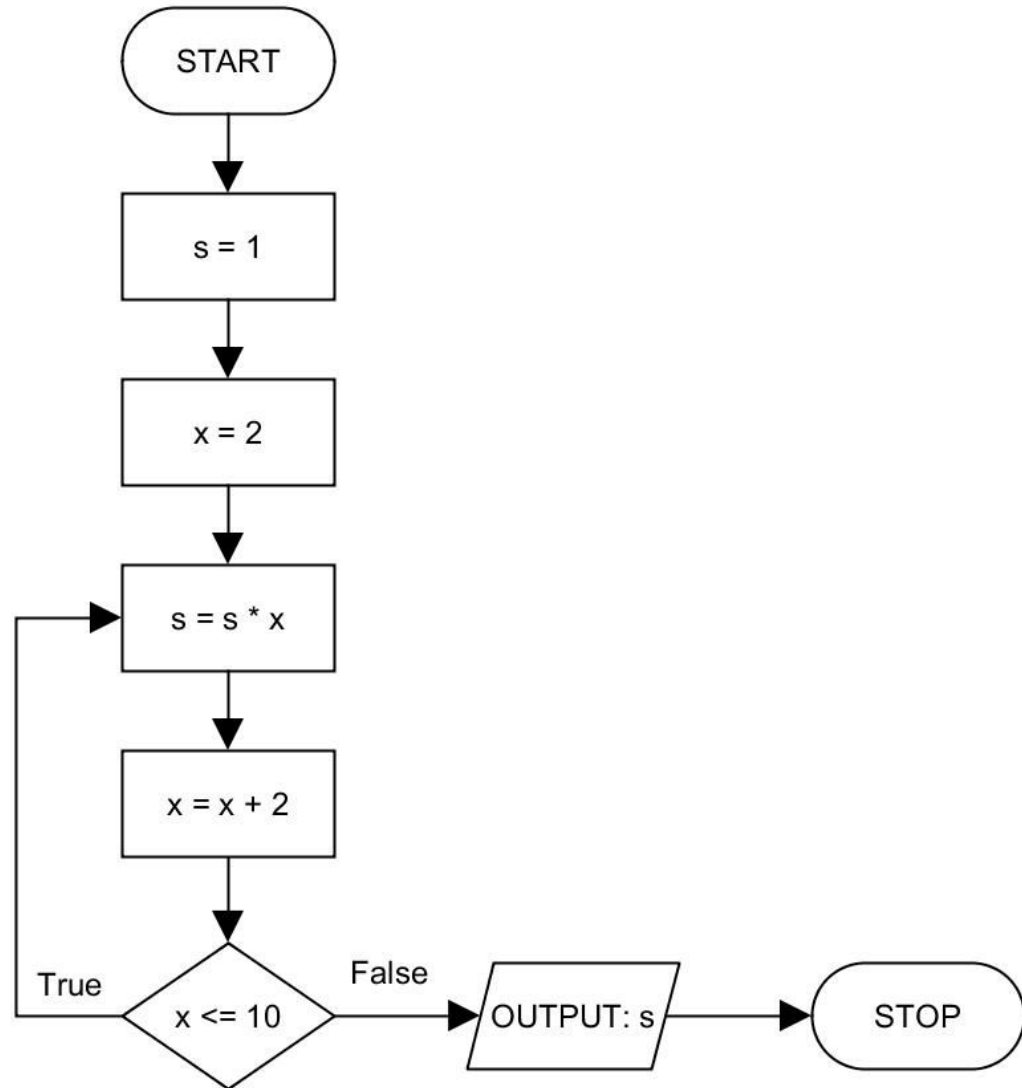
## Task 5:

Let us read an arbitrary real number (x) as an input datum, determine whether it is positive, negative or zero, and display the information.

# Designing and expressing algorithms

## Pseudocode and flowchart of the algorithm for task 2:

```
s = 1
x = 2
do
  s = s * x
  x = x + 2
while x <= 10
write s
```



# Designing and expressing algorithms

**Evaluation of the algorithm for task 2 in a tabular form:**

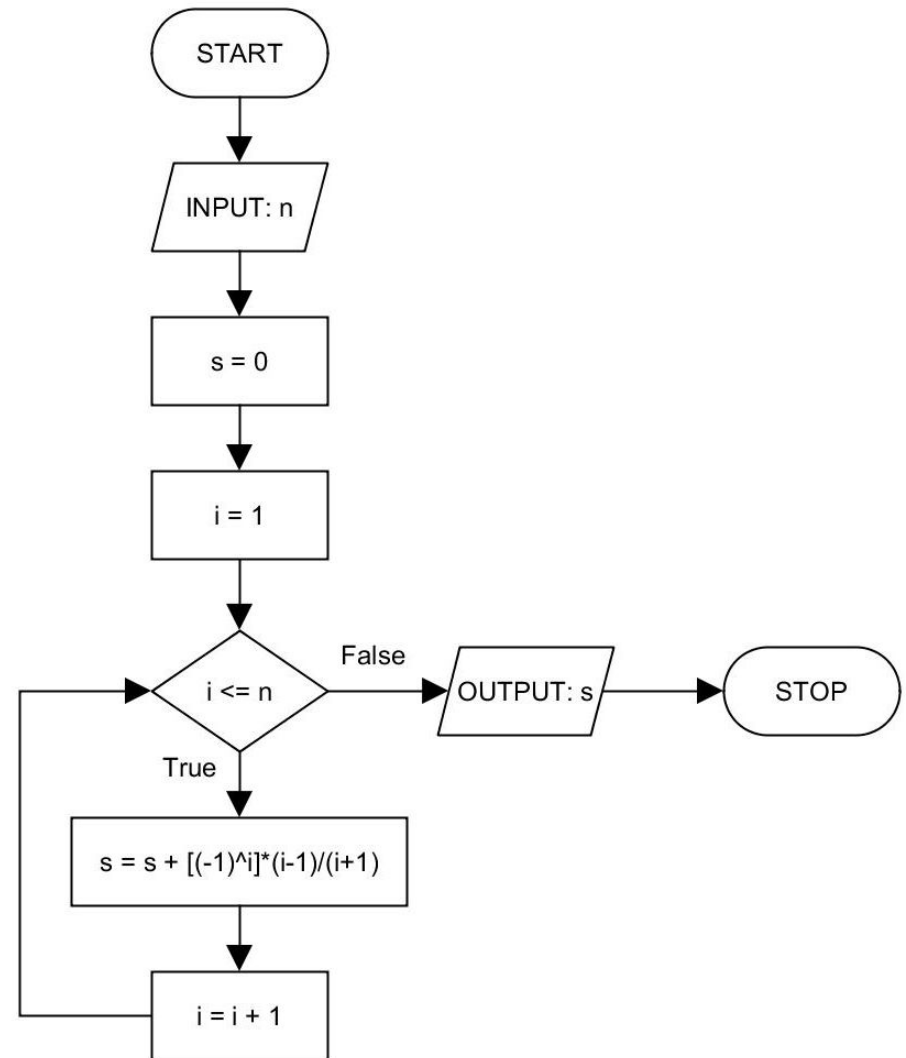
number of repetition	value of s	value of x
0	1	2
1	2	4
2	8	6
3	48	8
4	384	10
5	3840	12

# Designing and expressing algorithms

## Pseudocode and flowchart of the algorithm for task 3:

```
read n
s = 0
i = 1
while i <= n do
    s = s + [(-1)^i]*(i-1)/(i+1)
    i = i + 1
write s
```

```
read n
s = 0
for i=1 to n do
    s = s + [(-1)^i]*(i-1)/(i+1)
write s
```

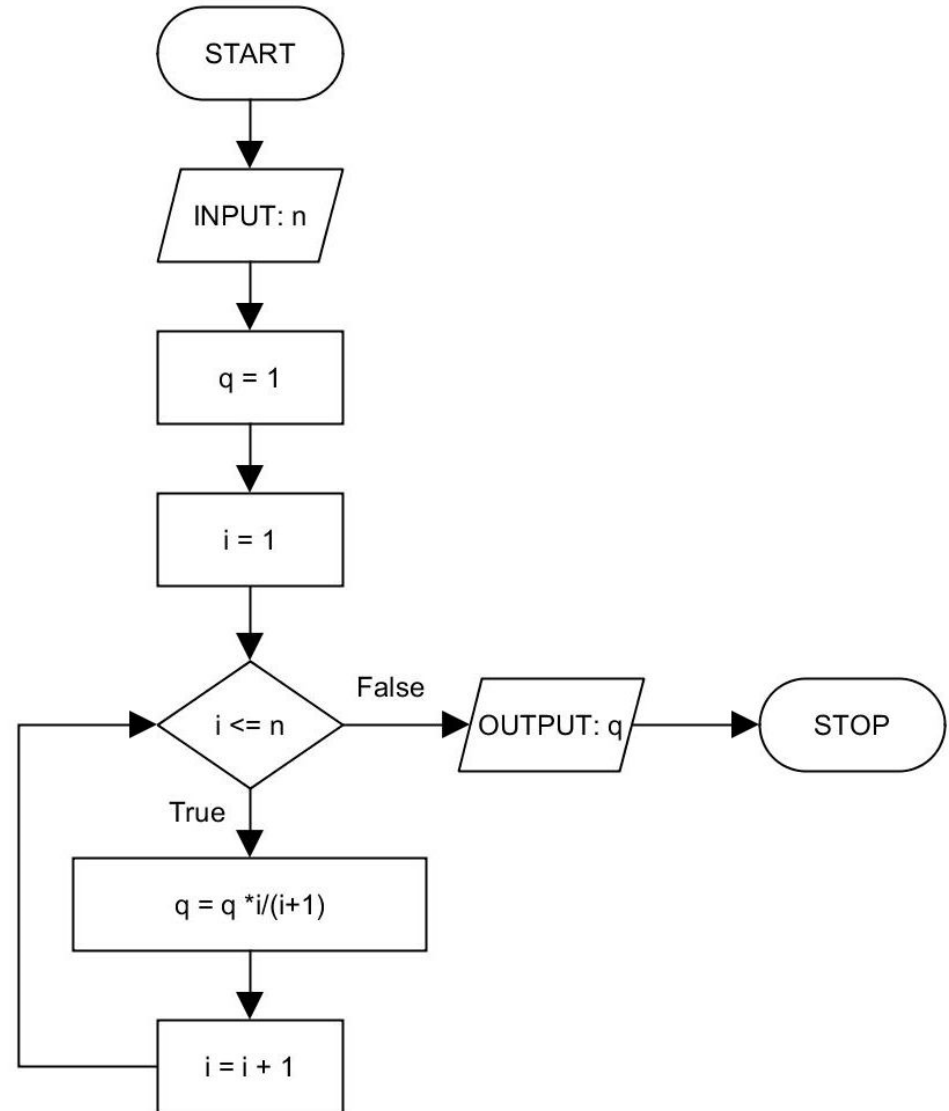


# Designing and expressing algorithms

## Pseudocode and flowchart of the algorithm for task 4:

```
read n
q = 1
i = 1
while i <= n do
    q = q * i / (i+1)
    i = i + 1
write q
```

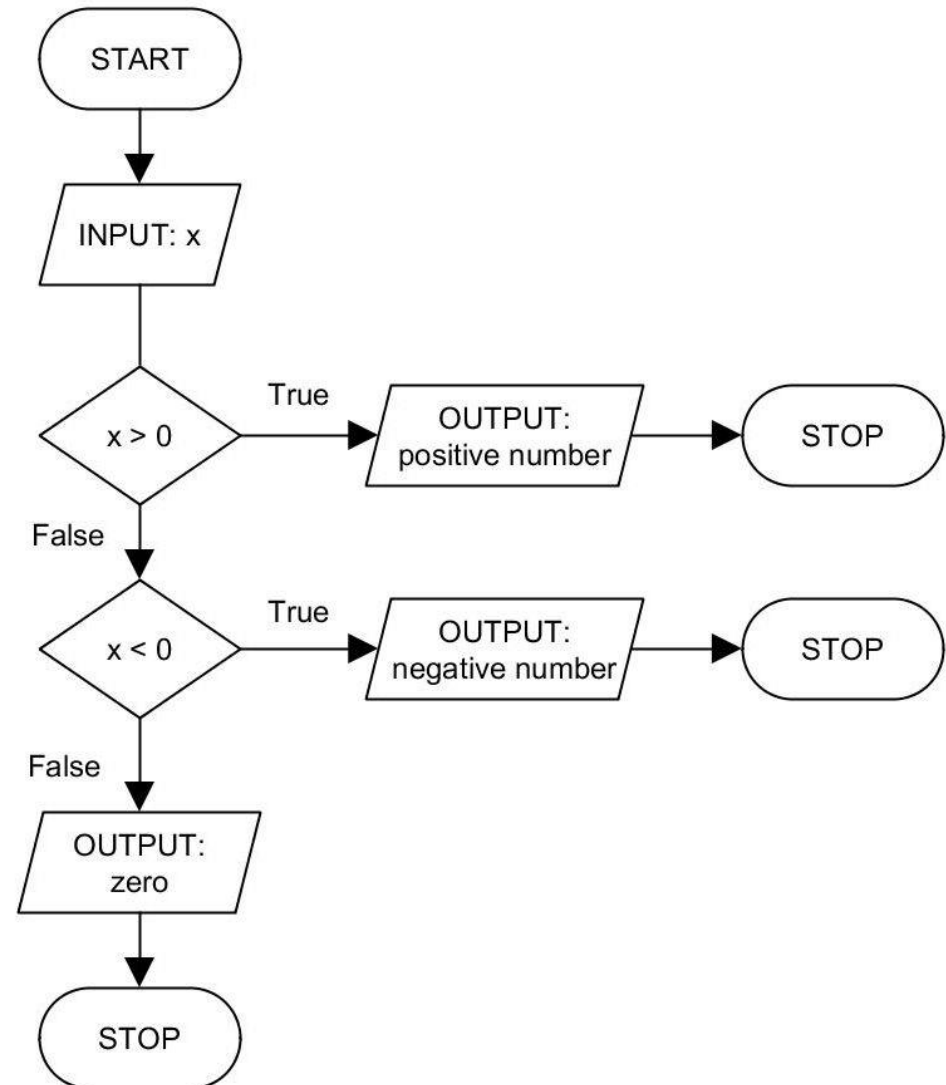
```
read n
q = 1
for i=1 to n do
    q = q * i / (i+1)
write q
```



# Designing and expressing algorithms

## Pseudocode and flowchart of the algorithm for task 5:

```
read x
if x > 0 then
    write „positive number”
else if x < 0 then
    write „negative number’
else
    write „zero”
```





# Implementation of algorithms in C programming language

**Source code of the algorithm for task 1 in C programming language (pre-test loop):**

```
#include <stdio.h>
```

```
main(){
```

```
    int s, x;
```

```
    s = 0;
```

```
    x = 1;
```

```
    while (x <= 10){
```

```
        s = s + x;
```

```
        x = x + 1;
```

```
    }
```

```
    printf(„The result: %d“, s);
```

```
}
```

# Implementation of algorithms in C programming language

**Source code of the algorithm for task 1 in C programming language (post-test loop):**

```
#include <stdio.h>

main(){
    int s, x;

    s = 0;
    x = 1;
    do{
        s = s + x;
        x = x + 1;
    } while (x <= 10);
    printf(„The result: %d“, s);
}
```