<texit info> author=Roman Putanowicz backgroundtext=http://www.L5.pk.edu.pl/~putanowr/iten.html title=Lab 5: More on control flow; Octave versus Matlab showbuttons=off </texit>
<texit>\input{/home/prac/putanowr/dokuwiki/data/media/wiki/latex/it_preamb.tex} \setcounter{chapter}{4} </texit> <texit>\begin{comment}</texit> **prev** | **up** | **next** <texit>\end{comment}</texit>

Lab 5 : More on control flow; Octave versus Matlab

Description

Loops with unspecified number of iterations. **While** loops . Difference between **while** and **do-until** loops. Transforming **for** loops into **while** loops. Calculating limits for sequences and series. The main differences between Octave and Matlab syntax. Running Matlab.

Skills to be acquired

- Understanding syntax and semantics of the “while” loops.
- Understanding the difference between loops with prescribed number of iterations and loops with unspecified number of iterations
- Knowing the basic differences between Octave and Matlab syntax.
- Knowing how to run Matlab under MS Windows.

Self study

The lab introduces the “while” statements for writing loops with unspecified number of iterations. Besides “for”, “if” and “while” statements Octave offers other flow control constructs that might be less frequently used but nevertheless worth knowing. They are the main topics for the self study:

- syntax and semantics of “do-until” loops,
- handling choices with “switch” statements.

Readings

- "Undetermined number of iteration loops"(PDF) – chapter 5 of dr A. Matuszak's handouts (other chapters available at http://www.l5.pk.edu.pl/~max/ti.html)
- "While loops" (PDF) – handouts for Information Techonology course in Polish (lecturer dr P.Mika)

Solved problems

Ex. 5.1.1

Implement a simple guessing game. The program picks randomly an integer value between 0 and N (where N is fixed in the code). The task of the user is to guess this number. The program asks the user until the right number is given.
Hint: The code below generates random integer x in the range [0,N]:

```
N = 100
x = floor(N * rand())
```

Solution.

Ex. 5.1.2

Starting from an arbitrary initial value x_0 the sequence \begin{equation*} x_{1+n} = \frac{1}{2}\left(x_n + \frac{A}{x_n}\right) \end{equation*} converges to te value $\sqrt{A}$. Write a script that allows the user to calculate such x_n that satisfies the condition \begin{equation*} \left|\sqrt{A} - x_n\right| < \varepsilon \end{equation*} for a user supplied values of A, x_0 and ε.

Solution.

Ex. 5.2.3

Write a program that calculates how man sides should have a regular polygon inscribed a circle of radious R so the relative error of approximating the circle area by the polygon area is less than 0.5%.

Solution

Questions and open problems

Ex. 5.2.1

Modify the guessing game from exercise 5.1.1 in such way that the program gives the user hints if the user's guess in smaller or greater than the picked number. What kind of strategy is possible with the modified game?

Ex. 5.2.2

Modify the program from exercise 5.2.1 in such way that the program counts the user's trials and displays the message when the game is finished. Introduce the restriction on number of user's trials.

Ex. 5.2.3

Modify the program from exercise 5.2.2 to allow a user to play several times. After the whole gaming session is finished the program should print a statistics of how many times the user was guessing in each single game.

Ex. 5.2.4

Write a simple console animation of a star character (*) that bounces forever in a line of 60 characters long.

Hint: check how string variables are handled in Octave. Check the documentation for `pause()` function.

Ex. 5.2.5

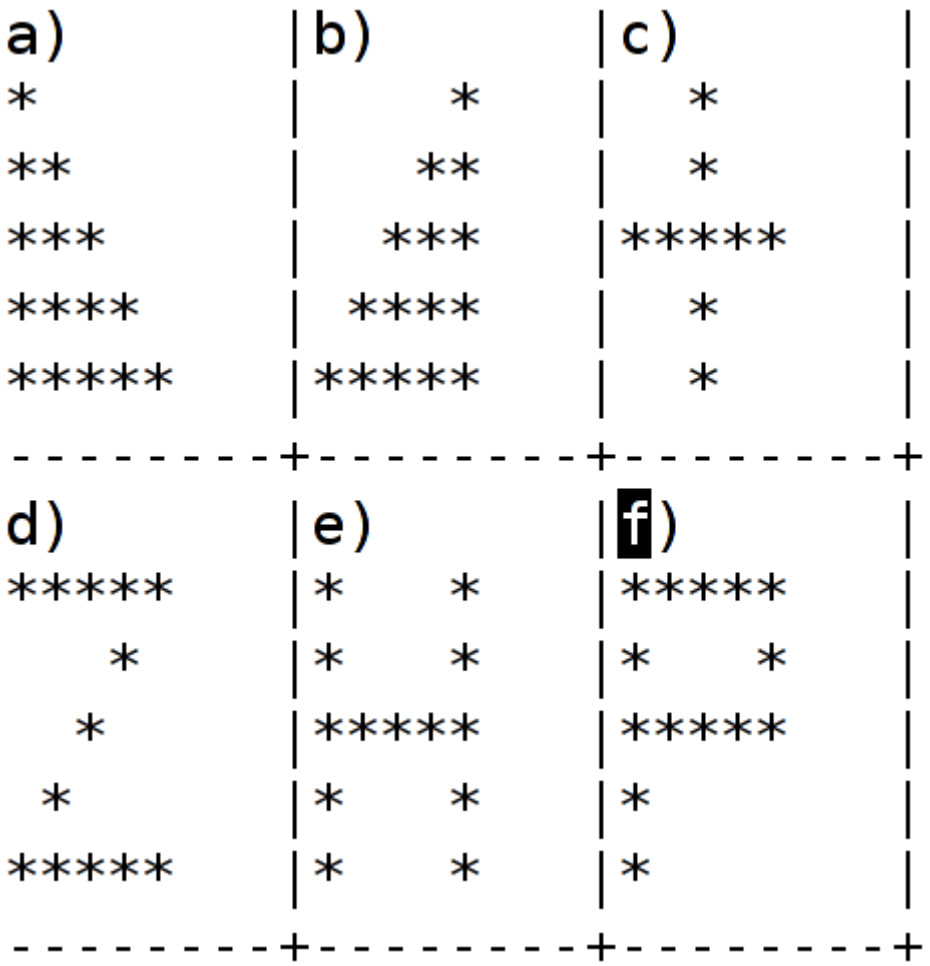
Given a parametric curve \begin{equation*} \begin{split} x(t)&= t^2 \cdot sin(2\pi t)\\ y(t)&= t \cdot cos(\pi t) \end{split} \end{equation*} for $t\in[0,2\pi]$, write a script that calculates the length of the curve by approximating it by linear segments. The program should allow the user to specify relative error for the length.

Ex. 5.2.6

Write a program that calculates how many elements a truncated Maclaurin series should have so the relative error of calculating the value of function f(x)=cos(x) for x=0.9 by this series is less than 2%.

Ex. 5.2.7

Write scripts that will print the asterix patterns shown in the figure below. Instead of using **for** loops use only **while** or **do-until** loops. Write the scipts so the user can specify the height of the pattern (as odd integer number). The patterns shown in the figure have height equal 5.



Ex. 5.2.8

Ex. 5.2.9

Links

- http://www.l5.pk.edu.pl/~pm/techn_inf.html – Another information technology course materials (in Polish)]]
- http://www.l5.pk.edu.pl/~max/ti.html – Yet another IT course (in Polish)

<texit>\begin{comment}</texit> **prev** | **up** | **next** <texit>\end{comment}</texit>