Seminar Objectives

- Inspection of documents (specification and design and code) of a given problem

Theoretical aspects

- Quality and Inspections.
- Defects.
- References: [Myers] - chapter 3; [Young] - chapter 18; [Frentiu] - chapter 4

[Myers] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004
[Young] M. Pezzand, M. Young, *Software Testing and Analysis: Process, Principles and Techniques*, John Wiley & Sons, 2008
[Frentiu] M. Frentiu, Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010

Assignment

Inspect the documents (problem statement, design, source code) for the received problem.

Inspection refers to the analysis and the highlighting of the current state of the documents into a report.

Inspection may cause modification of the analyzed documents, like:

- Clarification of the statement problem;
- Modification of the design and/or the source code.
- Use the same available documents from the Laboratory 1 assignments.
- For the identification of the ambiguities/defects the following *check-lists* will be used:
  - a. **Statement problem:** Lab01_Requirements Phase Defects Checklist.pdf;
  - b. **Design:** Lab01_Unit Design Phase Defects Checklist.pdf**;**
  - c. **Source code**: Lab01_Program Coding Phase Defects Checklist.pdf.
- For the inspected documents/artifacts a report will be realized (Lab01_Review Form.xls).

Write a program that reads natural numbers n1, n2, …, nk and prints the longest sequence ns, ns+1, …, nd, with 1≤s≤d≤k, that contains only prime numbers.
The program must have:
- a subalgorithm that reads the given numbers
- a function that verifies if a natural number is prime;
-a subalgorithm that computea the indexies s and d, 1≤s≤d≤k, with the property that ns, ns+1, …, nd are prime numbers;
- a subalgorithm that prints the numbers ns, ns+1, …, nd.

```
Program ProblemForInspection; {The longest sequence of prime numbers}
Type sirnat = array [1..100] of integer;
Var  N: sirnat; {array that contains the given numbers }
s,d, {the positions that give the requested sequence }
i, k: integer; {k=the length of array N}
Function Prim(m:integer):boolean; {True if  m is prime} Var i: integer; b: boolean;
Begin
b:=true; { "m is prime"}
if m<2 then b:=false else begin
          i:=1;
          while b and (i< m div 2) do
                    if m mod i = 0 then b:= false {m is not prime}
                    else b:=true; end {if};
          Prim := b
end ;
Procedure Secventa(i,k:integer; N:sirnat; var j:integer);
{i is given such that  N[i] is prime. We obtain the greatest j for witch the sequence N[i..j]
contains onloy prime numbers }
Begin j:=i;  While (j<=k) and Prim(N[j+1]) do j:=j+1; end;
Procedure LongSecv (k:integer; N:sirnat; s,d:integer);
{ We obtain the indexies s and d for witch the sequence  N[s..d] contains only prime numbers
and is the longest}
var i,j: integer;
Begin
i:=0; s:=0; d:=0; {the variable d is used for "the array does not contain prime numbers "}
While (i<k) do Begin
          while Prim(N[i]) and (i<k) do i:=i+1;
          if (i<k) then begin
                    Secventa (i,k,N,j);
                    if (j-i > d-s) then begin s:=i; d:=j end; {a longest sequence was found }
                    i:=j+2; {the position for a new sequence}
          end {if}
end {while}
```

Write a program that reads natural numbers n1, n2, …, nk and prints the longest sequence ns, ns+1, …, nd, with 1≤s≤d≤k, that contains only prime numbers.
The program must have:
- a subalgorithm that reads the given numbers
- a function that verifies if a natural number is prime;
- a subalgorithm that computes the position *start* and *length* sof the sequence with the property that any two concecutive numbers from the sequence nstart, …, nstart*length differs by a prime number;
- a subalgorithms that prints the numbers ns, ns+1, …, nd.

```
Program ProblemForInspoection;
Type sirnat = array [1..100] of integer;
Var N : sirnat; { array that contains the given numbers }
s,d, { {the positions that give the requested sequence }
i ,k : integer; {k= the length of array N }
Function Prim(m:integer):boolean; { True if  m is prime } Var i: integer; b: boolean;
Begin
b:=true; { "m is prime"}
if m<2 then b:=false else begin
          i:=1;
          while b and (i< m div 2) do
          if m mod i = 0 then b:= false {m is not prime}
          else b:=true; end {if};
Prim := b
end ;
procedure CeaMaiLungaSecv(nrE:integer;valE:sirnat;pozF,lungF:integer); {We obtain  start and length
for the requested sequence }
Begin
 pozF:=0; lungF:=0; pozI:=0; lungI:=0; i:= 0;
While i<=nrE do Begin
          If EstePrim(valE(i)-valE(i+1)) Then
                    If pozI=0 Then begin  pozI:= i; lungI:= 2;end
                    Else  lungI:=lungI+1;
          Else
                    If lungI>=lungF Then begin {a longest seqence was found}
                              lungF:= lungI;
                              pozF:= pozI;
                    end
          i:=i+1;
 end;
end;
Begin writeln('give the number of elements'); readln(k);
for i:=1 to k do readln(N[i]);
CeaMaiLungaSecv (k,N,s,d);
If (d=0) then writeln('No prime numbers')
else Begin writeln('The sequence is:'); for i:=s to d do write(N[i]:5) end
end.
```

Requirements Phase Defects Checklist

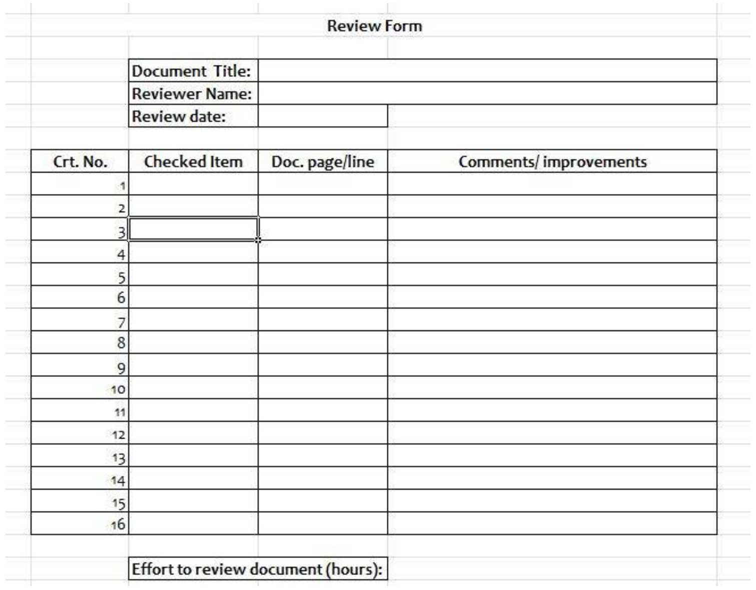
Sr.	Check Point / Defect Statement	Check Mark (√) the Appropriate Column	
		Yes	N/A
01	Requirements are incomplete.		
02	Requirements are missing.		
03	Requirements are incorrect.		
04	Initialization of the system state has not been considered.		
05	The functions have not been defined adequately.		
06	The user needs are inadequately stated.		
07	Comments		

Unit Design Phase Defects Checklist

Sr.	Check Point / Defect Statement	Check Mark (√) the Appropriate Column	
		Yes	N/A
01	Is the if-then-else construct used incorrectly?		
02	Is the dowhile construct used incorrectly?		
03	Are there infinite loops?		
04	Is the program readable?		
05	Is the program efficient?		
06	Is the algorithm expression too complicated?		
07	Is the nesting too deep?		
08	Is there any negative Boolean logic?		
09	Are there any compounded Boolean expressions?		
10	Is there any jumping in and out of loops?		

Coding Phase Defects Checklist

Sr.	Check Point / Defect Statement	Check Mark (√) the Appropriate Column	
		Yes	N/A
01	Decision logic is erroneous or inadequate.		
02	Branching is erroneous.		
03	There are undefined loop terminations.		
04	I/O format errors exist.		
05	Subprogram invocations are violated.		
06	There are errors in preparing or processing input data.		
07	Output processing errors exist.		
08	Error message processing errors exist.		
09	There is confusion in the use of parameters.		
10	There are errors in loop counters.		
11	Errors are made in writing out variable names.		
12	Variable type and dimensions are incorrectly declared.		



Review Form

Document Title:	
Reviewer Name:	
Review date:	

Crt. No.	Checked Item	Doc. page/line	Comments/ improvements
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Effort to review document (hours):	