Carmichaelzahlen und Pseudoprimzahlen in MuPAD

- ```
  isCarmichael:=proc(n)local a;
  begin
    if n<2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    for a from 2 to n - 1 do
       if igcd(a, n) = 1 and a^(n-1) mod n <> 1 then return(FALSE); end_if
    end_for;
    return(TRUE);
  end_proc:
  ```

- ```
  isPseudoprime:=proc(n,a)
  begin
    if igcd(2, n) = 2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    if a^(n-1) mod n = 1 then return(TRUE); end_if;
    return(FALSE);
  end_proc:
  ```

- ```
  isPseudoprime1:=proc(n)    local a;
  begin
    if igcd(2, n) = 2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    for a from 2 to n - 2 do
       if a^(n-1) mod n = 1 then return(TRUE, a); end_if
    end_for;
    return(FALSE);
  end_proc:
  ```

- ```
  CarmichaelListe:=proc(Anfang, Ende) local i, Liste;
  begin
    Liste:=[];
    for i from Anfang to Ende do
       if isCarmichael(i) = TRUE then Liste:=append(Liste, i); end_if;
    end_for;
    return(Liste);
  end_proc:
  ```

- ```
  PseudoprimeListe:=proc(a,Anfang,Ende)  local i, Liste;
  begin
    Liste:=[];
    for i from Anfang to Ende do
       if isPseudoprime(i,a) = TRUE then Liste:=append(Liste, i); end_if;
    end_for;
    return(Liste);
  end_proc:
  ```

- ```
  psp2:=PseudoprimeListe(2,1,5000);
  psp3:=PseudoprimeListe(3,1,5000);
  psp5:=PseudoprimeListe(5,1,5000);
  ```
- ```
  Menge_psp2:={op(psp2)};
  Menge_psp3:={op(psp3)};
  Menge_psp5:={op(psp5)};
  ```
- ```
  Menge_psp2 intersect Menge_psp3 intersect Menge_psp5;
  ```

Carmichaelzahlen und Pseudoprimzahlen in MuPAD

- ```
  isCarmichael:=proc(n)  local a;
  begin
    if n<2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    for a from 2 to n - 1 do
     if igcd(a, n) = 1 and a^(n-1) mod n <> 1 then
        return(FALSE);
     end_if
    end_for;
    return(TRUE);
  end_proc:
  ```

- ```
  isPseudoprime:=proc(n,a)
  begin
    if igcd(2, n) = 2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    if a^(n-1) mod n = 1 then return(TRUE); end_if;
    return(FALSE);
  end_proc:
  ```

- ```
  isPseudoprime1:=proc(n)    local a;
  begin
    if igcd(2, n) = 2 then return(FALSE); end_if;
    if isprime(n) = TRUE then return(FALSE); end_if;
    for a from 2 to n - 2 do
       if a^(n-1) mod n = 1 then return(TRUE, a); end_if
    end_for;
    return(FALSE);
  end_proc:
  ```
  ```

- ```
  CarmichaelListe:=proc(Anfang, Ende) local i, Liste;
  begin
    Liste:=[];
    for i from Anfang to Ende do
       if isCarmichael(i) = TRUE then
          Liste:=append(Liste, i);
       end_if;
    end_for;
    return(Liste);
  end_proc:
  ```

- ```
  PseudoprimeListe:=proc(a,Anfang,Ende)local i, Liste;
  begin
    Liste:=[];
    for i from Anfang to Ende do
       if isPseudoprime(i,a) = TRUE then
          Liste:=append(Liste, i);
       end_if;
    end_for;
    return(Liste);
  end_proc:
  ```

- ```
  psp2:=PseudoprimeListe(2,1,5000);
  psp3:=PseudoprimeListe(3,1,5000);
  psp5:=PseudoprimeListe(5,1,5000);
  ```
- ```
  Menge_psp2:={op(psp2)};
  Menge_psp3:={op(psp3)};
  Menge_psp5:={op(psp5)};
  ```
- ```
  Menge_psp2 intersect Menge_psp3 intersect Menge_psp5;
  ```
  ```