Exercice 1 :

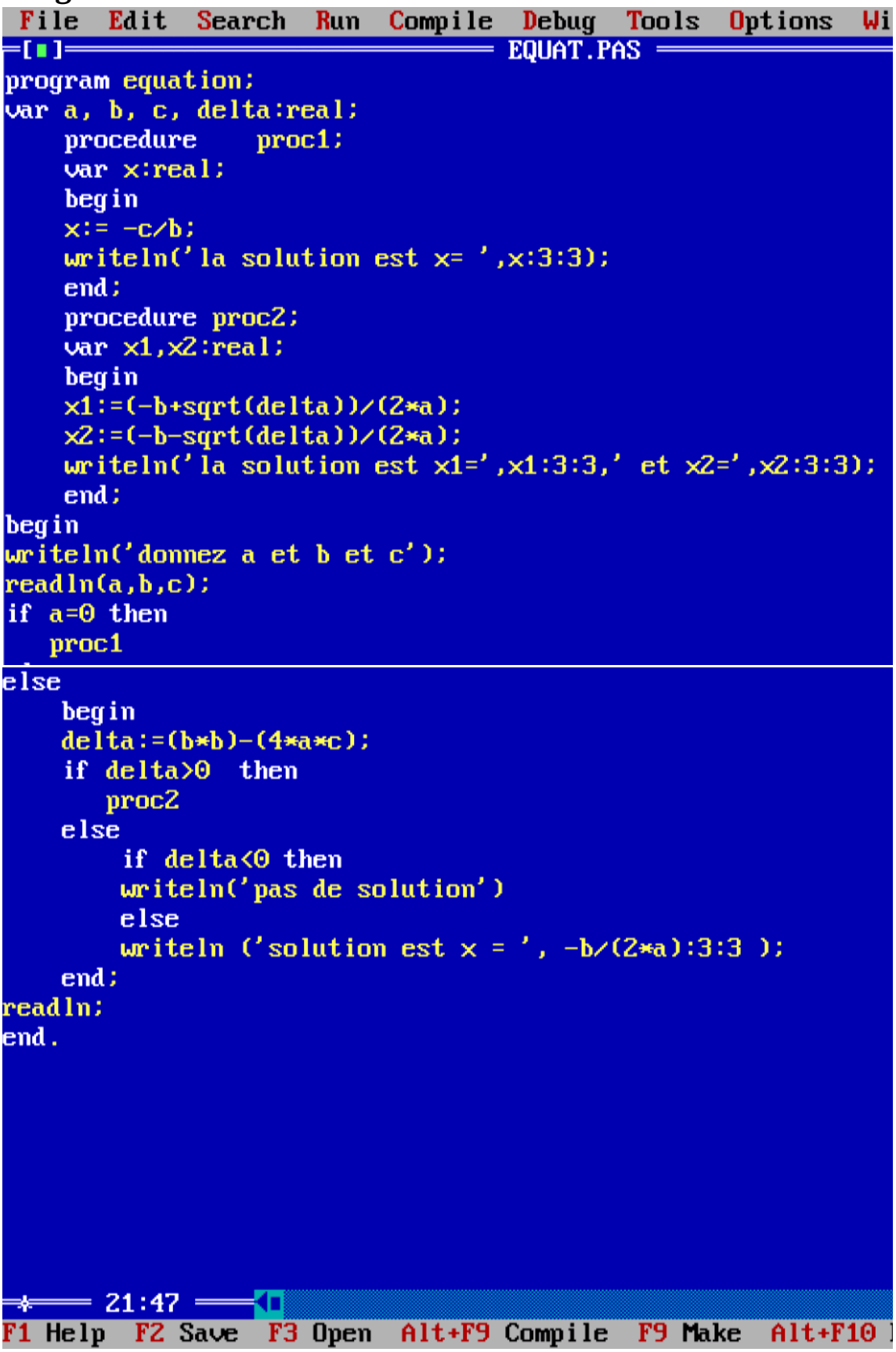
Ecrire un programme pascal qui permet de trouver la solution de l'équation Ax2 + bx + c = 0 en traitant tous les cas possible (a = 0 ; b = 0 ; delta =0 , …)

Le programme doit être sous forme de procédures où chaque procédure traite un cas

Algorithme

```
Algorithme exo1
Var    a, b, c delta : réel

Procédure proc1
Var x : réel
Début
        x← -c/b
        Ecrire ('la solution est x= ', x)
Fin

Procédure proc2
Var x1, x2 : réel
Début
        x1← -b-sqrt(delta)/2*a
        x1← -b+sqrt(delta)/2*a
        Ecrire ('la solution est x1= ', x1, ' et x2= ', x2)
Fin

Debut
    Ecrire ('entrez les coefficients a, b et c')
    Lire (a, b, c)
        Si a=0 Alors
            Proc1
        Sinon
            Delta← (b*b)-(4*a*c)
            Si delta >0 Alors
                Proc2
            Sinon
                Si delta <0 Alors
                    Ecrire ('pas de solution')
                Sinon
                    Ecrire ('la solution est x= ', -b/(2*a))
                Finsi
            Finsi
        Finsi
Fin
```

Programme


File Edit Search Run Compile Debug Tools Options Wi
EQUAT.PAS
program equation;
var a, b, c, delta:real;
procedure proc1;
var x:real;
begin
x:= -c/b;
writeln('la solution est x= ',x:3:3);
end;
procedure proc2;
var x1,x2:real;
begin
x1:=(-b+sqrt(delta))/(2*a);
x2:=(-b-sqrt(delta))/(2*a);
writeln('la solution est x1=',x1:3:3,' et x2=',x2:3:3);
end;
begin
writeln('donnez a et b et c');
readln(a,b,c);
if a=0 then
proc1
else
begin
delta:=(b*b)-(4*a*c);
if delta>0 then
proc2
else
if delta<0 then
writeln('pas de solution')
else
writeln ('solution est x = ', -b/(2*a):3:3);
end;
readln;
end.
21:47
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10

Exercice 2 :

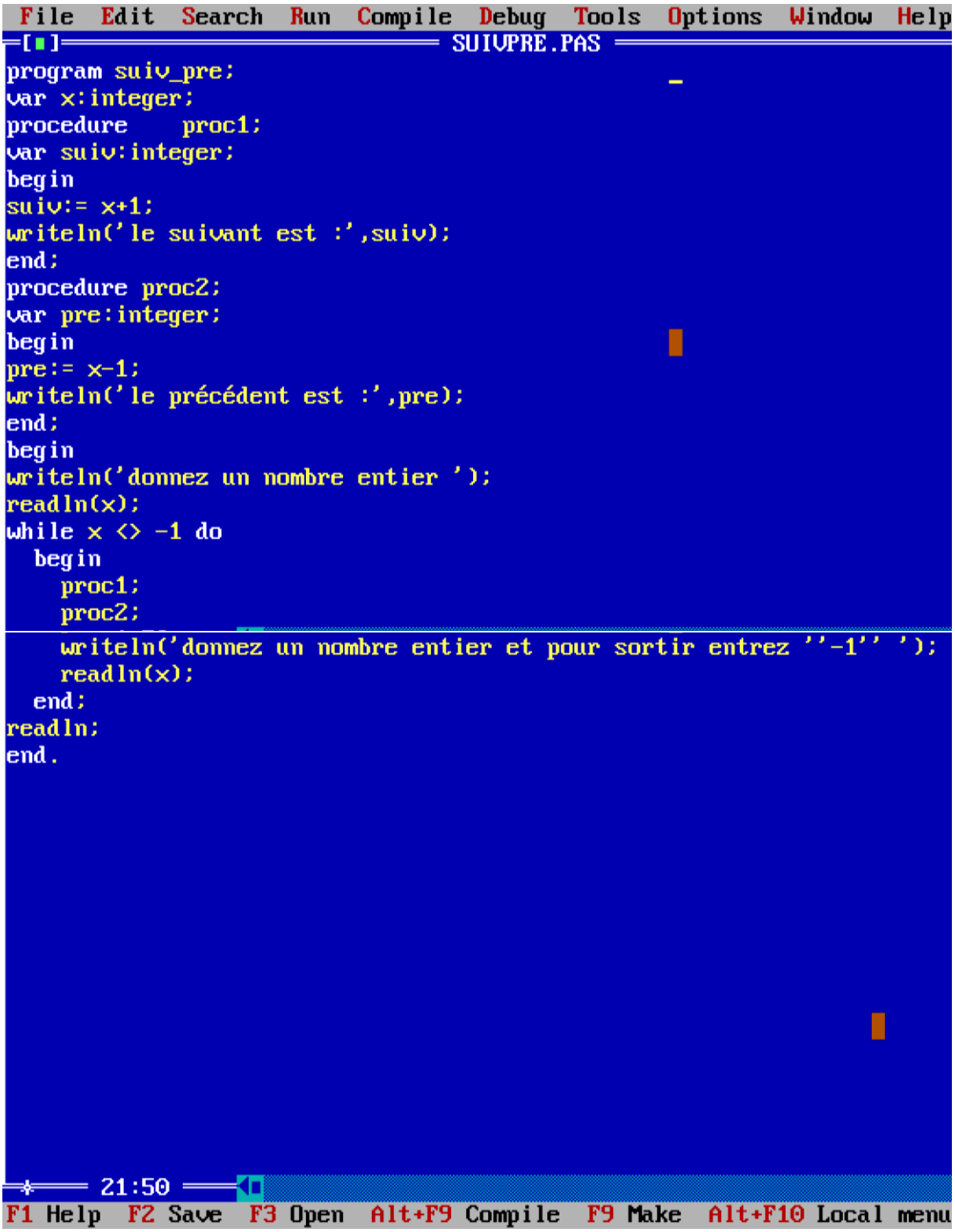
Sans utiliser succ(x) ni pred(x), Ecrire un programme pascal qui permet d'afficher le successeur et le prédécesseur d'un entier positif introduit au clavier, le programme tourne indéfiniment jusqu'en tapant (-1).

Le programme doit être sous forme d'un programme principale et deux procédures l'une pour le successeur et l'autre pour le prédécesseur.

Algorithme

```
Algorithme exo2
Var    x : entier

Procédure proc1
Var suiv : entier
Début
    suiv← x+1
    Ecrire ('le suivant est : ', x)
Fin

Procédure proc2
Var pre : entier
Début
    pre ← x-1
    Ecrire ('le précédent est : ', pre)
Fin

Debut
  Ecrire ('Entrez un nombre entier')
  Lire (x)
    TQ  x≠-1 faire
      Proc1
      Proc2
    FinTQ
Fin
```

Programme


File Edit Search Run Compile Debug Tools Options Window Help
SUIVPRE.PAS
program suiv_pre;
var x:integer;
procedure proc1;
var suiv:integer;
begin
suiv:= x+1;
writeln('le suivant est :',suiv);
end;
procedure proc2;
var pre:integer;
begin
pre:= x-1;
writeln('le précédent est :',pre);
end;
begin
writeln('donnez un nombre entier ');
readln(x);
while x <> -1 do
begin
proc1;
proc2;
writeln('donnez un nombre entier et pour sortir entrez ''-1'' ');
readln(x);
end;
readln;
end.
21:50
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Exercice 3 :

a) Ecrire la fonction Turbo-pascal factorielle, qui à un entier positif n associe n!.

b) Intégrer cette fonction dans un programme qui demande deux entiers n et k ≤ n à l'utilisateur, et qui renvoie

$$\binom{n}{k} \; tel\ que \; \binom{n}{k} = \frac{n!}{(n-k)! * (k!)}$$

Algorithme

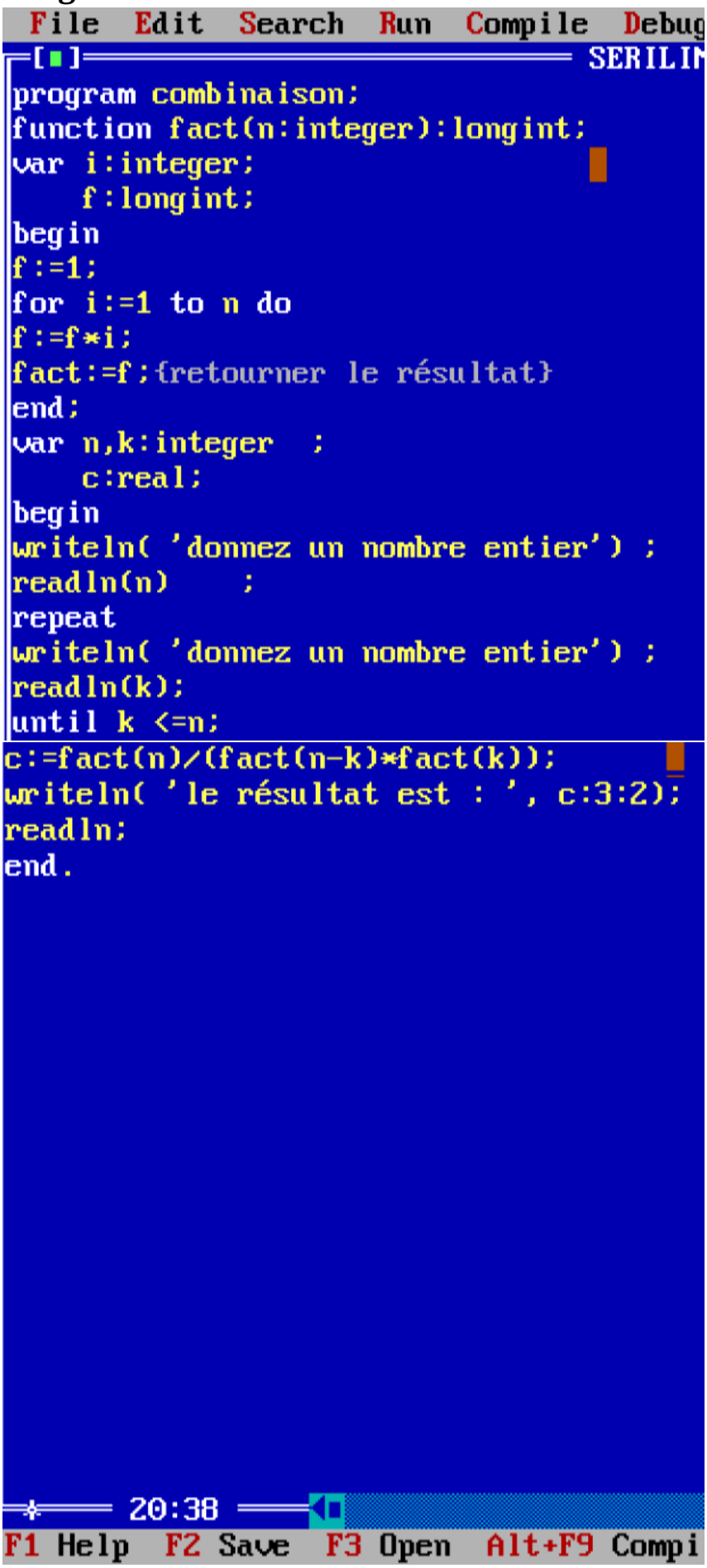
a)

```
Fonction fact (n:entier) : entier
Var  i, f : entier
Debut
f←1;
   Pour i de 1 jusqu'à n faire
      f← f*i
   Finpour
fact←f
Fin
```

b)

```
Algorithme exo3
Fonction fact (n : entier) : entier
Var  i, f : entier
Debut
f←1;
   Pour i de 1 jusqu'à n faire
      f← f*i
   Finpour
fact←f
Fin

Var    n, k : entier
       c : reel
 Debut
   Ecrire ('Entrez un nombre entier n')
   Lire (n)
   Répéter
      Ecrire ('Entrez un nombre entier k')
      Lire (k)
   Jusqu'à k<=n
   c← fact(n) / (fact(n-k)*fact(k))
   ecrire ('le résultat est : ',x)
Fin
```

Programme



```
File  Edit  Search  Run  Compile  Debug
[■]                                SERILIN
program combinaison;
function fact(n:integer):longint;
var i:integer;
    f:longint;
begin
f:=1;
for i:=1 to n do
f:=f*i;
fact:=f;{retourner le résultat}
end;
var n,k:integer  ;
    c:real;
begin
writeln( 'donnez un nombre entier') ;
readln(n)    ;
repeat
writeln( 'donnez un nombre entier') ;
readln(k);
until k <=n;
c:=fact(n)/(fact(n-k)*fact(k));
writeln( 'le résultat est : ', c:3:2);
readln;
end.
20:38
F1 Help  F2 Save  F3 Open  Alt+F9 Compi
```

Exercice 4 :

Ecrire un programme pascal qui fait calculer $(x + a)^n$ selon la formule suivante (x, a et n sont des valeurs saisis au clavier)

$$(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}$$

Le programme doit être sous forme d'un programme principale et deux fonctions.

Algorithme

```
Algorithme s2_exo4

Fonction puiss (x:reel ; n:entier) : reel
Var  i : entier
       p : reel
Debut
 p:=1;
   Pour i de 1 jusqu'à n faire
       p← p*x
   Finpour
 Puiss←p
Fin

Fonction fact (n : entier) : entier
Var  i, f : entier
Debut
f←1;
   Pour i de 1 jusqu'à n faire
       f← f*i
   Finpour
fact←f
Fin

Fonction comb(n, k : entier) : reel
 Début
       c← fact(n) /( fact(n-k)* fact(k))
 Fin

Var     n, k : entier
        x, a : reel
 Debut
   Ecrire ('Entrez un nombre entier')
   Lire (n)
   Ecrire ('Entrez un nombre x ')
   Lire (x)
   Ecrire ('Entrez un nombre a')
   Lire (a)
    s ←0;
   Pour k de 0 jusqu'a n faire
       s← s +(comb(n,k)* puiss(x, k) * puiss(a, n-k))
   Finpour
   Ecrire ('le résultat est : ', s)
 Fin
```

Programme

File Edit Search Run Compile Debug Tools Options Win

S2_EXO4.PAS

```
program s2_exo4;
function puiss(x:real;n:integer):real;
var i:integer;
    p:real;
begin
p:=1;
for i:=1 to n do
p:=p*x;  {writeln(p); }
puiss:=p;{retourner le résultat}
end;

function fact(n:integer):longint;
var i:integer;
    f:longint;
begin
f:=1;
for i:=1 to n do
f:=f*i;
fact:=f;{retourner le résultat}
end;

function comb(x,y:integer):real;
begin
{writeln(fact(x)/(fact(x-y)*fact(y)));}
comb:=fact(x)/(fact(x-y)*fact(y));{retourner le résultat}
end;

var n,k:integer;
    s,x,a:real;
begin
writeln( 'donnez un nombre entier') ;
readln(n);
writeln( 'donnez un nombre reel x') ;
readln(x);
writeln( 'donnez un nombre reel a') ;
readln(a) ;
s:=0;
for k:=0 to n do
s:=s+(comb(n,k)*puiss(x,k) *puiss(a,n-k));
writeln( 'le résultat est : ', s:3:2);
readln;
end.
```

21:17

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 L

Exercice 5 :

Ecrire un programme pascal qui calcule e^x selon le développement de Taylor

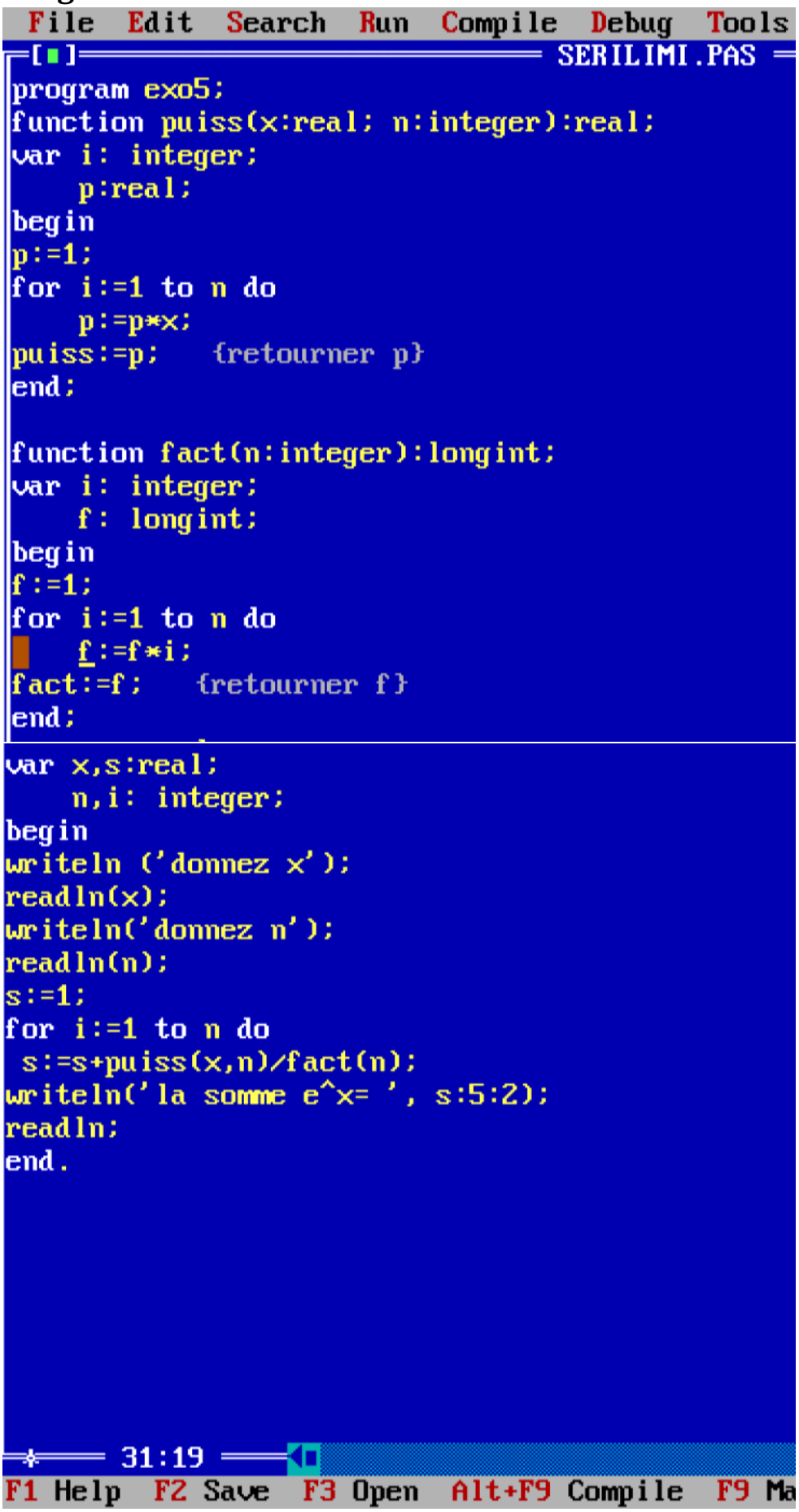
$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots + \frac{x^n}{n!} \quad , \qquad -\infty < x < +\infty$$

n et x sont deux valeurs saisis au clavier .

Algorithme

```
Algorithme exo5

Fonction puiss (x:reel ; n:entier) : reel
Var  i : entier
       p : reel
Debut
 p:=1;
   Pour i de 1 jusqu'à n faire
       p← p*x
   Finpour
 Puiss←p
Fin

Fonction fact (n : entier) : entier
Var  i , f: entier
Debut
f←1;
   Pour i de 1 jusqu'à n faire
       f← f*i
   Finpour
fact←f
Fin

Var    x, s : reel
         n: entier

 Debut
   Ecrire ('Entrez un nombre ')
   Lire (x)
   Ecrire ('Entrez un nombre ')
   Lire (n)
    s ←1;
   Pour i de 1 jusqu'a n faire
       s← s + puiss(x, n) / fact(n)
   Finpour
  Ecrire ('la somme  e^x =  ', s)
 Fin
```

Programme


File Edit Search Run Compile Debug Tools
SERILIMI.PAS
program exo5;
function puiss(x:real; n:integer):real;
var i: integer;
p:real;
begin
p:=1;
for i:=1 to n do
p:=p*x;
puiss:=p; {retourner p}
end;
function fact(n:integer):longint;
var i: integer;
f: longint;
begin
f:=1;
for i:=1 to n do
f:=f*i;
fact:=f; {retourner f}
end;
var x,s:real;
n,i: integer;
begin
writeln ('donnez x');
readln(x);
writeln('donnez n');
readln(n);
s:=1;
for i:=1 to n do
s:=s+puiss(x,n)/fact(n);
writeln('la somme e^x= ', s:5:2);
readln;
end.
31:19
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Ma