

I Dans le programme suivant, compléter la fonction `nouv` qui rend l'adresse d'un nouveau chaînon contenant les deux valeurs passées en argument.

II Qu'écrit le programme ?

```
procedure h(n,a,b,c:integer);
begin
  writeln('':3*(4-n), 'h(',n,',',a,',',b,',',c,') début');
  if n>1 then
    begin
      h(n-1,b,a,c+1);
      h(n-2,a,c,b);
      h(n-1,c,a,b)
    end;
  writeln('':3*(4-n), 'h(',n,',',a,',',b,',',c,') fin')
end;
type liste=^chainon;
      chainon=record val:integer; suite:liste end;
function nouv(val:integer;suite:liste):liste;
begin
  (* à compléter *)
end;
procedure aff(l:liste);
begin
  while l<>nil do with l^ do begin write(val,' ');l:=suite end;
  writeln
end;
function inter(a,b:liste):liste;
begin
  if a=nil then a:=b
    else a^.suite:=inter(b,a^.suite);
  inter:=a
end;
```

```

const n=40;
      d:array[1..5] of integer=( , , , , );
var t:array[1..n] of integer;
      a,b:liste;
      i,j:integer;
begin
  for i:=1 to 5 do write(d[i], ' ');
  writeln;
  h(3,10*d[1],10*d[2],10*d[3]);
  for i:=1 to n do t[i]:=i;
  for i:=2 to n do if t[i]=i then
    for j:=2 to n div i do if odd(j) then t[i*j]:=i+j;
  for i:=1 to n do write(t[i], ' ');
  writeln;
  a:=nil;
  b:=nil;
  for i:=1 to 5 do a:=nouv( d[i],a);
  for i:=1 to 5 do b:=nouv(11-d[i],b);
  aff(a);aff(b);aff(inter(a,b));aff(a);aff(b)
end.

```

III Combien de lignes afficherait l'instruction
 $h(10, 3*d[5], 3*d[4], 3*d[3])$?

IV Combien de fois la comparaison $t[i]=i$ est-elle faite ?

Combien de fois $n \text{ div } i$ est-il évalué ?

Combien de fois $\text{odd}(j)$ est-il évalué ?

Combien de fois $t[i*j]:=i+j$ est-il exécuté ?

V Répondez de nouveau aux quatre questions précédentes en supposant que n est quelconque et grand. On donnera les réponses sous une forme simple comme $\Theta(n^3)$.

VI Quelles seraient les cinq dernières lignes écrites par le programme si on en remplaçait la ligne

```
      else a^.suite:=inter(b,a^.suite);
```

par

```
      else a^.suite:=inter(a^.suite,b);
```

VII Même question en remplaçant cette ligne par les deux lignes :

```
      else if (b=nil) or (a^.val<b^.val) then a:=inter(b,a)
```

```
      else a^.suite:=inter(b,a^.suite);
```

Barème

Toutes les questions valent 2 points sauf le II dans lequel **h** vaut 6,66 points, **t** vaut 6 points et les listes chaînées valent 4 points.

Cela fait un total de 28,66 points.

Barème détaillé

h : 8,66 points

II : 20 lignes : $20 \times 0.33 = 6,66$ points

III : **h**(10, ...) écrit 10088 lignes : 2 points (1 point si le nombre **n**'est pas trop faux)

t : 10 points

II : 40 nombres : 6 points, -1 point par nombre faux

	IV	V	
t [i]= i	39	$\Theta(n)$	
n div i	16	$\Theta(n/\ln n)$	$8 \times 0,5 = 4$ points
odd(j)	62	$\Theta(n \ln \ln n)$	
t [i * j]:= i + j	28	$\Theta(n \ln \ln n)$	

listes chaînées 10 points

I : 0,5 pour **new**(**a**), 0,5 pour **nouv**:=**a**, 0,5 pour **a**^.**val**:=**val**,
0,5 pour **a**^.**suite**:=**suite**, -0,5 par erreur, total 2 points

II 5 lignes 1 1 1 0,5 0,5

VI 5 lignes 0 0 1 0,5 0,5

VII 5 lignes 0 0 1 0,5 0,5

Corrigé

III : Si **h**(**n**, ...) écrit l_n lignes, alors $l_0 = l_1 = 2$ et $l_n = 2 + 2l_{n-1} + l_{n-2}$.

t : **t**[**i**] vaut **i** quand **i** est premier ou égal à une puissance de 2 (1, 2, 4, 8, 16 ou 32). Sinon si **i** est impair, **t**[**i**] vaut $p + i/p$ où **p** est le plus grand diviseur premier de **i**. Et si **i** est pair, **t**[**i**] vaut $2^k + i/2^k$ où 2^k est la plus grande puissance de 2 qui divise **i**.

listes chaînées : Pour le II, **inter** intercale les deux listes.

Pour le VI, **inter** colle **b** au bout de **a**.

Pour le VII, si **a** et **b** étaient triées, alors **inter** fusionnerait ces deux listes en une seule liste triée.

```

    Le programme :
procedure h(n,a,b,c:integer);
begin
  writeln('':3*(4-n), 'h(',n,',',a,',',b,',',c,') début');
  if n>1 then
  begin
    h(n-1,b,a,c+1);
    h(n-2,a,c,b);
    h(n-1,c,a,b)
  end;
  writeln('':3*(4-n), 'h(',n,',',a,',',b,',',c,') fin')
end;
type liste=^chainon;
  chainon=record val:integer; suite:liste end;
function nouv(val:integer;suite:liste):liste;
begin new(nouv); nouv^.val:=val; nouv^.suite:=suite end;
procedure aff(l:liste);
begin
  while l<>nil do with l^ do begin write(val,' ');l:=suite end;
  writeln
end;
var k:integer;
function inter(a,b:liste):liste;
begin
  if a=nil then a:=b else
  if (k=3) and ((b=nil) or (a^.val<b^.val)) then a:=inter(b,a)
    else if k=1 then a^.suite:=inter(b,a^.suite)
      else a^.suite:=inter(a^.suite,b);

  inter:=a
end;
var c:array[0..10] of integer;
function z(i:integer):integer; begin inc(c[i]); z:=0 end;
const n=40;
  d:array[1..5] of integer=(5,7,1,9,3);
var t:array[1..n] of integer;
  a,b:liste;
  i,j:integer;

```

```

begin
  for i:=1 to 5 do write(d[i], ' ');
  writeln;
  h(3,10*d[1],10*d[2],10*d[3]);
  for i:=1 to 4 do c[i]:=0;
  for i:=1 to n do t[i]:=i;
  for i:=2 to n do if t[i]=i+z(1) then
    for j:=2 to n div i+z(2) do if odd(j+z(3)) then
      t[i*j]:=i+j+z(4);
  for i:=1 to n do write(t[i], ' ');
  for k:=1 to 3 do
  begin
    writeln;
    a:=nil;
    b:=nil;
    for i:=1 to 5 do a:=nouv( d[i],a);
    for i:=1 to 5 do b:=nouv(11-d[i],b);
    aff(a);aff(b);aff(inter(a,b));aff(a);aff(b)
  end;
  writeln('t[i]=i ',c[1], ' n div i ', c[2],
    ' odd(j) ',c[3], ' t[i*j]:=i+j ',c[4]);
  c[0]:=2; c[1]:=2;
  for i:=2 to 10 do c[i]:=2+2*c[i-1]+c[i-2];
  for i:=0 to 10 do write(i:4,':',c[i]);
  writeln
end.

```

écrit:

5 7 1 9 3

```

h(3,50,70,10) début
  h(2,70,50,11) début
    h(1,50,70,12) début
      h(1,50,70,12) fin
        h(0,70,11,50) début
          h(0,70,11,50) fin
            h(1,11,70,50) début
              h(1,11,70,50) fin
            h(2,70,50,11) fin
          fin
        fin
      fin
    fin
  fin
end

```

```

    h(1,50,10,70) début
    h(1,50,10,70) fin
  h(2,10,50,70) début
    h(1,50,10,71) début
    h(1,50,10,71) fin
      h(0,10,70,50) début
      h(0,10,70,50) fin
    h(1,70,10,50) début
    h(1,70,10,50) fin
  h(2,10,50,70) fin
h(3,50,70,10) fin
1 2 3 4 5 5 7 8 6 7 11 7 13 9 8 16 17 11 19 9 10 13 23 11 10
15 12 11 29 17 31 32 14 19 12 13 37 21 16 13
3 9 1 7 5
8 2 10 4 6
3 8 9 2 1 10 7 4 5 6
3 8 9 2 1 10 7 4 5 6
8 9 2 1 10 7 4 5 6

3 9 1 7 5
8 2 10 4 6
3 9 1 7 5 8 2 10 4 6
3 9 1 7 5 8 2 10 4 6
8 2 10 4 6

3 9 1 7 5
8 2 10 4 6
8 3 9 2 10 4 6 1 7 5
3 9 2 10 4 6 1 7 5
8 3 9 2 10 4 6 1 7 5
t[i]=i 39   n div i 16   odd(j) 62   t[i*j]:=i+j 28
  0:2   1:2   2:8   3:20   4:50   5:122   6:296   7:716
  8:1730   9:4178   10:10088

```