

QUICK-R

```

100 MODE 2:STACK,&7000,&1000 >XZ
110 DEFINT b-s:DEFREAL a,t,x:n=100:DIM a(n),tri(n):b=b: >RJ
d=d:g=g:f=f
120 ' >RD
130 'remplissage d'un tableau de nombres au hasard >RE
140 FOR i=1 TO n:a(i)=RND*n:NEXT >AL
150 ' >RG
160 'transmission au tableau de tri >RH
170 !GIVE,n*5,@a(1):!GET,n*5,@tri(1) >FM
180 ' >RK
190 'indication des bornes Gauche et Droite >TA
200 g=1:d=n:!GIVE,2,@g,2,@d >WY
210 ' >RD
220 'tri du tableau >RE
230 z=TIME:GOSUB 370:z=(TIME-z)/300 >EH
240 ' >RG
250 'recuperation du tableau trie >RH
260 !GIVE,n*5,@tri(1):!GET,n*5,@a(1) >FM
270 ' >RK
280 'affichage du tableau trie et du temps de tri >TA
290 FOR i=1 TO n:PRINT a(i):NEXT >AE
300 PRINT z >ZG
310 END >RH
320 ' >RF
330 '----- >RG
340 'Routine "QUICKSORT" d'apres C.A.R. Hoare >RH
350 ' >RJ
360 'definition des variables locales >RK
370 !LOCAL,2,@g,2,@d,2,@f >UJ
380 'recuperation des bornes de tri >TB
390 !GET,2,@g,2,@d >NQ
400 ' >RE
410 'tri fini si 1 seul element >RF
420 IF g>=d THEN 700 >ML
430 ' >RH
440 'tri par rapport a l'element central, echange avec >RJ
l'element de gauche
450 i=(g+d)/2:x=tri(i):tri(i)=tri(g) >FN
460 ' >TA
470 'on tri de droite a gauche >TB
480 f=g:i=d >ZA
490 ' >TD
500 'trier le tas >RF
510 WHILE i>f >MK
520 ' >RH
530 ' placer a gauche tout element inferieur >RJ
540 IF tri(i)<x THEN f=f+1:!GIVE,5,@tri(i),5,@tri(f): >YW
!GET,5,@tri(f),5,@tri(i): GOTO 580
550 ' passer a l'element suivant >TA
560 i=i-1 >EH
570 ' >TC
580 WEND >DB
590 ' >TE
600 'placer element de partage entre les tas >RG
610 tri(g)=tri(f):tri(f)=x >WQ
620 ' >RJ

```

```

630 'trier le tas de gauche >RK
640 b=f-1:!GIVE,2,@g,2,@b:GOSUB 370 >DM
650 ' >TB
660 'trier le tas de droite >TC
670 b=f+1:!GIVE,2,@b,2,@d:GOSUB 370 >DK
680 ' >TE
690 'restauration des variables locales >TF
700 !ENDLOCAL,2,@g,2,@d,2,@f >XB
710 ' >RJ
720 RETURN >ZE

```

QUICK-S

```

100 MODE 2:STACK,&7000,&1000 >XZ
110 DEFINT b-w:DEFSTR a,t,x:n=100:DIM a(n):b=b:d=d:g=g: >FG
f=f
120 ' >RD
130 'remplissage d'un tableau de chaines >RE
140 PRINT"creee chaines..." >ZJ
150 FOR i=1 TO n:FOR j=1 TO (1+RND*6):a(i)=a(i)+CHR$(65 >TC
+RND*7):NEXT j,i
160 ' >RH
170 'transmission au tableau de tri >RJ
180 DIM tri(n):FOR i=1 TO n:tri(i)=a(i):NEXT:b=FRE("") >ZE
190 ' >TA
200 'indication des bornes Gauche et Droite >RC
210 g=1:d=n:!GIVE,2,@g,2,@d >WZ
220 ' >RE
230 'tri du tableau >RF
240 PRINT"trie...":z=TIME:GOSUB 380:z=(TIME-z)/300 >YP
250 ' >RH
260 'recuperation du tableau trie >RJ
270 FOR i=1 TO n:a(i)=tri(i):NEXT:ERASE tri:b=FRE("") >YW
280 ' >TA
290 'affichage du tableau trie et du temps de tri >TB
300 FOR i=1 TO n:PRINT a(i):NEXT >AW
310 PRINT z >ZH
320 END >RJ
330 ' >RG
340 '----- >RH
350 'Routine "QUICKSORT" d'apres C.A.R. Hoare >RJ
360 ' >RK
370 'definition des variables locales >TA
380 !LOCAL,2,@g,2,@d,2,@f >UK
390 'recuperation des bornes de tri >TC
400 !GET,2,@g,2,@d >NG
410 ' >RF
420 'tri fini si 1 seul element >RG
430 IF g>=d THEN 710 >MN
440 ' >RJ
450 'tri par rapport a l'element central, echange avec >RK
l'element de gauche
460 i=(g+d)/2:x=tri(i):tri(i)=tri(g) >FP
470 ' >TB
480 'on tri de droite a gauche >TC
490 f=g:i=d >ZB
500 ' >RF

```

```

510 'trier le tas
520 WHILE i>f
530 '
540 ' placer a gauche tout element inferieur
550 IF tri(i)<x THEN f=f+1:GIVE,3,@tri(i),3,@tri(f):
:GET,3,@tri(f),3,@tri(i): GOTO 590
560 ' passer a l'element suivant
570 i=i-1
580 '
590 WEND
600 '
610 'placer element de partage entre les tas
620 tri(g)=tri(f):tri(f)=x
630 '
640 'trier le tas de gauche
650 b=f-1:GIVE,2,@g,2,@b:GOSUB 380
660 '
670 'trier le tas de droite
680 b=f+1:GIVE,2,@b,2,@d:GOSUB 380
690 '
700 'restauration des variables locales
710 :ENDLOCAL,2,@g,2,@d,2,@f
720 '
730 RETURN

```

SHELL-R

```

100 MEMORY &8F00
110 MODE 2:STACK,&9000,&1000
120 DEFINT a-s:DEFREAL t-z:n=400:DIM xtab(n),noeud(n,3)
:filsg=0:n=0:ptr=0
130 '
140 'pointeurs du noeud
150 ptrx=0:occ=1:filsg=2:filsd=3
160 '
170 'initialisation de l'arbre
180 racine=1:id=1
190 '
200 'remplissage d'arbre
210 PRINT"insere dans arbre..."
220 FOR i=0 TO 100
230 x=ROUND(100*RND,0):GIVE,2,@racine:GOSUB 350
240 NEXT
250 '
260 'affichage de l'arbre
270 :GIVE,2,@racine:GOSUB 540
280 '
290 END
300 '
310 '-----
320 'Routine de tri binaire
330 '
340 'recuperation du pointeur de noeud
350 :GET,2,@ptr
360 '
370 'si noeud vide alors ajouter valeur
380 IF noeud(ptr,ptrx)=0 THEN noeud(ptr,ptrx)=id:xtab(i

```

```

>RG ● d)=x:noeud(ptr,occ)=1:GOTO 490
>NA ● 390 ' >TC
>RJ ● 400 'identite entre valeur X et celle du noeud >RE
>RK ● 410 IF x=xtab(noeud(ptr,ptrx)) THEN noeud(ptr,occ)=noeu >GC
>YP ● d(ptr,occ)+1:GOTO 490
420 ' >RG
>TB ● 430 'aller a gauche >RH
>EJ ● 440 IF x<xtab(noeud(ptr,ptrx)) THEN :GIVE,2,@ptr,2,@fil >TT
>TD ● sg:GOSUB 740:GIVE,2,@noeud(ptr,filsg):GOSUB 350:GOTO 4
90 >DC ●
>RG ● 450 ' >RK
>RH ● 460 'aller a droite >TA
>WR ● 470 :GIVE,2,@ptr,2,@filsg:GOSUB 740:GIVE,2,@noeud(ptr, >BY
>RK ● filsd):GOSUB 350
>TA ● 480 ' >TC
>DP ● 490 RETURN >ZJ
>TC ● 500 ' >RF
>TD ● 510 '----- >RG
>DM ● 520 'Affichage d'arbre >RH
>TF ● 530 ' >RJ
>RH ● 540 :LOCAL,2,@ptr >NQ
>XC ● 550 ' >TA
>RK ● 560 :GET,2,@ptr >LY
>ZF ● 570 ' >TC
● 580 IF noeud(ptr,occ)=0 THEN 670 >ZE
590 ' >TE
600 'affiche fils gauche >RG
● 610 :GIVE,2,@noeud(ptr,filsg):GOSUB 540 >KM
620 PRINT DEC$(noeud(ptr,occ),"###");" => ";xtab(noeud >RA
(ptr,ptrx))
>HK ● 630 ' >RK
>XC ● 640 'affiche fils droit >TA
>AG ● 650 :GIVE,2,@noeud(ptr,filsg):GOSUB 540 >KN
>RE ● 660 ' >TC
>RF ● 670 :ENDLOCAL,2,@ptr >RP
>CM ● 680 ' >TE
>RH ● 690 RETURN >AA
700 ' >RH
>RJ ● 710 '----- >RJ
>MK ● 720 'Allocation de fils a un noeud "n" >RK
>TA ● 730 ' >TA
>RC ● 740 :LOCAL,2,@n,2,@filsg >TG
>FG ● 750 ' >TC
>LB ● 760 :GET,2,@n,2,@filsg >RT
>VP ● 770 ' >TE
>EB ● 780 IF noeud(n,filsg)=0 THEN id=id+1:noeud(n,filsg)=id >YB
>RH ● 790 ' >TG
>RJ ● 800 :ENDLOCAL,2,@n,2,@filsg >WT
>ZN ● 810 ' >RK
>TA ● 820 RETURN >ZF

```

SHELL-S

```

100 MEMORY &8F00:OPENOUT"x":MEMORY HIMEM-1:CLOSEOUT >AE
110 MODE 2:STACK,&9000,&1000 >XC
120 DEFINT a-l,n-z:DEFSTR m:n=600:DIM mottab(n),noeud(n >FN
,3):filsg=0:n=0:ptr=0

```

130 ' >RE ●	650 ' >TB
140 'pointeurs du noeud >RF	660 'affiche fils gauche >TC
150 ptrmot=0:occ=1:filsg=2:filsd=3 >EH	670 !GIVE,2,@noeud(ptr,filsg):GOSUB 600 >KQ
160 ' >RH ●	680 PRINT DEC\$(noeud(ptr,occ),"###");" => ";mottab(noeud(ptr,ptrmot)) >YH
170 'initialisation de l'arbre >RJ	690 ' >TF
180 racine=1:id=1 >MK	700 'affiche fils droit >RH
190 ' >TA ●	710 !GIVE,2,@noeud(ptr,filsd):GOSUB 600 >KG
200 'remplissage d'arbre >RC	720 ' >RK
210 OPEN IN"article3":WHILE NOT EOF >FY	730 !ENDLOCAL,2,@ptr >RL
220 'lire et formater ligne >RE ●	740 ' >TB
230 INPUT#9,ligne\$:GOSUB 930:i=INSTR(ligne\$," ") >UE	750 RETURN >ZH
240 ' trier mot >RG	760 ' >TD
250 WHILE i<>0 >VB ●	770 '----- >TE
260 mot=LOWER\$(MID\$(ligne\$,1,i-1)):ligne%=MID\$(ligne\$,i+1) >GX	780 'Allocation de fils a un noeud "n" >TF
270 IF mot<>" THEN PRINT"insere ";mot:!GIVE,2,@racine >TU ●	790 ' >TG
:GOSUB 410 >TD	800 !LOCAL,2,@n,2,@fils >TD
280 i=INSTR(ligne\$," ") >DD ●	810 ' >RK
290 WEND >MV	820 !GET,2,@n,2,@fils >RP
300 WEND:CLOSE IN >RE	830 ' >TB
310 ' >RF ●	840 IF noeud(n,fils)=0 THEN id=id+1:noeud(n,fils)=id >YY
320 'affichage de l'arbre >ZG	850 ' >TD
330 !GIVE,2,@racine:GOSUB 600 >RH	860 !ENDLOCAL,2,@n,2,@fils >WZ
340 ' >TB ●	870 ' >TF
350 END >TB	880 RETURN >AB
360 ' >RK	890 '----- >TH
370 '----- >TA	900 'suppression espaces superflus >RK
380 'Routine de tri binaire >TB ●	910 ' >TA
390 ' >TC	920 'supprime signes de ponctuation & co >TB
400 'recuperation du pointeur de noeud >RE	930 ligne\$=" "+ligne\$+" ":s\$="!?\$%&'()*_+-. /:;@[]"+ >WD
410 !GET,2,@ptr >LR ●	CHR\$(34):FOR i=2 TO LEN(ligne\$)-1:j=INSTR(s\$,MID\$(ligne\$,i,1)):IF j<>0 THEN ligne%=LEFT\$(ligne\$,i-1)+" "+MID\$(ligne\$,i+1)
420 ' >RG	940 NEXT >EJ
430 'si noeud vide alors ajouter mot >RH	950 ' >TE
440 IF mottab(noeud(ptr,ptrmot))="" THEN noeud(ptr,ptrmot)=id:mottab(id)=mot:noeud(ptr,occ)=1:GOTO 550 >BD ●	960 'supprime espaces a gauche >TF
450 ' >RK	970 i=1:k\$=" ":WHILE k\$=" ":k\$=LEFT\$(ligne\$,1):IF k\$=" " AND ligne\$=" " THEN ligne\$="":RETURN ELSE IF k\$=" " THEN ligne%=MID\$(ligne\$,2) >VA
460 'identite entre mot et mot du noeud >TA ●	980 WEND >DF
470 IF mot=mottab(noeud(ptr,ptrmot)) THEN noeud(ptr,occ)=noeud(ptr,occ)+1:GOTO 550 >TN	990 ' >TJ
480 ' >TC ●	1000 'supprime espaces a droite >XD
490 'aller a gauche >TD	1010 i=LEN(ligne\$):k\$=" ":WHILE k\$=" ":k\$=RIGHT\$(ligne\$,1):IF k\$=" " AND ligne\$=" " THEN ligne\$="":RETURN ELSE IF k\$=" " THEN ligne%=LEFT\$(ligne\$,LEN(ligne\$)-1) >AN
500 IF mot<mottab(noeud(ptr,ptrmot)) THEN !GIVE,2,@ptr,2,@filsg:GOSUB 800:!GIVE,2,@noeud(ptr,filsg):GOSUB 410:GOTO 550 >EA ●	1020 WEND >HH
510 ' >RG	1030 ' >XG
520 'aller a droite >RH ●	1040 'supprime autres espaces superflus >XH
530 !GIVE,2,@ptr,2,@filsd:GOSUB 800:!GIVE,2,@noeud(ptr,filsd):GOSUB 410 >BN	1050 i=INSTR(ligne\$," "):WHILE i<>0:j=i:WHILE MID\$(ligne\$,j,1)=" ":j=j+1:WEND:IF j>i+1 THEN ligne%=LEFT\$(ligne\$,i)+MID\$(ligne\$,j) >WR
540 ' >RK ●	1060 j=i:i=i+INSTR(MID\$(ligne\$,i+1)," "):IF i=j THEN i=0 >AE
550 RETURN >ZF	1070 WEND >JC
560 ' >TB	1080 ' >YB
570 '----- >TC ●	1090 'ajoute un espace en fin >YC
580 'Affichage d'arbre >TD	1100 ligne%=ligne\$+" " >QG
590 ' >TE	1110 ' >XF
600 !LOCAL,2,@ptr >NM ●	1120 RETURN >EH
610 ' >RH	
620 !GET,2,@ptr >LV	
630 ' >RK ●	
640 IF noeud(ptr,occ)=0 THEN 730 >ZY	