281 You are given a list variable L assigned a nonempty list. All changes to L must be via procedure $swap$, defined as

$$swap \ = \ \langle i, j \colon \Box L \cdot L := i \rightarrow L\,j \mid j \rightarrow L\,i \mid L \rangle$$

(a) Write a program to reassign L a new list obtained by rotating the old list one place to the right (the last item of the old list is the first item of the new).

(b) (rotate) Given an integer r, write a program to reassign L a new list obtained by rotating the old list r places to the right. (If $r<0$, rotation is to the left $-r$ places.) Recursive execution time must be at most $\#L$.

(c) (segment swap) Given an index p, swap the initial segment up to p with the final segment beginning at p.

After trying the question, scroll down to the solution.

(a) Write a program to reassign L a new list obtained by rotating the old list one place to the right (the last item of the old list is the first item of the new).

§ Let $R = L' = L[\#L-1;\ 0;..\#L-1]$. Let $Q = L' = L[i;\ 0;..i;\ i+1;..\#L]$. Then

$R \Leftarrow i:= \#L-1.\ Q$

$Q \Leftarrow \textbf{if } i=0 \textbf{ then } ok \textbf{ else } swap\ (i-1)\ i.\ i:= i-1.\ Q \textbf{ fi}$

Proof of R refinement:

	$i:= \#L-1.\ Q$	expand Q and substitution
$=$	$L' = L[\#L-1;\ 0;..\#L-1;\ \#L-1+1;..\#L]$	final segment is empty
$=$	$L' = L[\#L-1;\ 0;..\#L-1]$	
$=$	R	

Proof of Q refinement, first case:

	$i=0 \wedge ok$	expand ok
$=$	$i=0 \wedge L'=L \wedge i'=i$	
$\Rightarrow$	$L' = L[i;\ 0;..i;\ i+1;..\#L]$	
$=$	Q	

Proof of Q refinement, last case:

	$i\neq0 \wedge (swap\ (i-1)\ i.\ i:= i-1.\ Q)$	expand $swap$ and Q
$=$	$i\neq0 \wedge (L:= (i-1) \to L\ i \mid i \to L(i-1) \mid L.\ i:= i-1.\ L' = L[i;\ 0;..i;\ i+1;..\#L])$	subst twice
$=$	$i\neq0 \wedge L' = ((i-1) \to L\ i \mid i \to L(i-1) \mid L)[i-1;\ 0;..i-1;\ i;..\#L]$	divide final segment
$=$	$i\neq0 \wedge L' = ((i-1) \to L\ i \mid i \to L(i-1) \mid L)[i-1;\ 0;..i-1;\ i;\ i+1;..\#L]$	index
$=$	$i\neq0 \wedge L' = L[i;\ 0;..i-1;\ i-1;\ i+1;..\#L]$	combine middle two segments
$\Rightarrow$	$L' = L[i;\ 0;..i;\ i+1;..\#L]$	
$=$	Q	

Now for the timing.

$t' = t+\#L-1 \Leftarrow i:= \#L-1.\ t' = t+i$

$t' = t+i \Leftarrow \textbf{if } i=0 \textbf{ then } ok \textbf{ else } swap\ (i-1)\ i.\ i:= i-1.\ t:= t+1.\ t' = t+i \textbf{ fi}$

Proof of first refinement:

	$i:= \#L-1.\ t' = t+i$	substitution
$=$	$t' = t+\#L-1$	

Proof of last refinement, first case:

	$i=0 \wedge ok$	expand ok
$=$	$i=0 \wedge L'=L \wedge i'=i \wedge t'=t$	
$\Rightarrow$	$t' = t+i$	

Proof of last refinement, last case:

	$i\neq0 \wedge (swap\ (i-1)\ i.\ i:= i-1.\ t:= t+1.\ t' = t+i)$	substitution law three times
$=$	$i\neq0 \wedge t' = t+1+i-1$	simplify and specialization
$=$	$t' = t+i$	

This should be just right for a **for**-loop. Let M be a list constant equal to the original value of L . Since all changes to L are via $swap$, $\#M=\#L$ always. Let invariant

$A\ i = L = M[0;..\#M-i\ ;\ \#M-1;\ \#M-i;..\#M-1]$

Then $A\ 1 = L=M$ and $A'(\#L)=R$.

$R \Leftarrow \textbf{for } i:= 1;..\#L \textbf{ do } swap\ (\#L-i-1)\ (\#L-i) \textbf{ od}$

(b) (rotate) Given an integer r , write a program to reassign L a new list obtained by rotating the old list r places to the right. (If $r<0$, rotation is to the left $-r$ places.) Recursive execution time must be at most $\#L$.

no solution given

(c) (segment swap) Given an index p, swap the initial segment up to p with the final segment beginning at p.

§ The problem can be stated as $0 \le p < \#L \Rightarrow L' = L[p;..\#L \; ; \; 0;..p]$. We introduce variables a and b for the lengths of the left and right segments. During execution, the extreme parts of the list $L[0;..p-a]$ and $L[p+b;..\#L]$ will be in place, with the center portions $L[p-a;..p]$ and $L[p;..p+b]$ still to be swapped. Define

$$Q \;=\; 0<b \Rightarrow L' = L[0;..p-a \; ; \; p;..p+b \; ; \; p-a;..p \; ; \; p+b;..\#L]$$

Now refine:

$$0 \le p < \#L \Rightarrow L' = L[p;..\#L \; ; \; 0;..p] \;\Leftarrow\; a{:=}\,p.\; b{:=}\,\#L-p.\; Q$$

$Q \;\Leftarrow\;$ **if** $a=0$ **then** ok
else if $a<b$
then $L{:=}\,L[0;..p-a \; ; \; p+b-a;..p+b \; ; \; p;..p+b-a \; ; \; p-a;..p \; ; \; p+b;..\#L]$.
$b{:=}\,b-a.\; Q$
else $L{:=}\,L[0;..p-a \; ; \; p;..p+b \; ; \; p-a+b;..p \; ; \; p-a;..p-a+b \; ; \; p+b;..\#L]$.
$a{:=}\,a-b.\; Q$ **fi fi**

The two assignments to L are not allowed, so we must still refine them. But they swap segments of equal size, so they are easier than the original problem.

$L{:=}\,L[0;..p-a \; ; \; p+b-a;..p+b \; ; \; p;..p+b-a \; ; \; p-a;..p \; ; \; p+b;..\#L] \;\Leftarrow$
for $i{:=}\,p-a;..p$ **do** $swap\ i\ (i+b)$ **od**

$L{:=}\,L[0;..p-a \; ; \; p;..p+b \; ; \; p-a+b;..p \; ; \; p-a;..p-a+b \; ; \; p+b;..\#L] \;\Leftarrow$
for $i{:=}\,p;..p+b$ **do** $swap\ i\ (i-a)$ **od**

The time for the whole problem is $\#L$, and the time for Q is $a+b$.