```
var numbers[100] ;

function siftDown(root, bottom) {
   var done ; var maxChild ; var temp ;
   done = false ;
   while (root * 2 <= bottom and not done) do {
     if (root * 2 == bottom) then {
       maxChild = root * 2
     }
     else if (numbers[root * 2] > numbers[root * 2 + 1]) then { maxChild = root * 2 }
          else { maxChild = root * 2 + 1 }
     if (numbers[root] < numbers[maxChild]) then {
       temp = numbers[root] ;
       numbers[root] = numbers[maxChild] ;
       numbers[maxChild] = temp ;
       root = maxChild
     }
     else { done = true }
   }
   return 0
}

function heapSort(size) {
  var temp ; var i ;
  i = (size / 2) - 1 ;
  while (i >= 0) do {
     call siftDown(i, size - 1) ;
     i = i - 1
  }
  i = size - 1 ;
  while (i >= 1 ) do {
   temp = numbers[0] ;
    numbers[0] = numbers[i] ;
    numbers[i] = temp ;
    call siftDown(0, i - 1) ;
    i = i - 1
  }
  return 0
}

function main() {
     var x ;
     x = read() ;
     for i = 0 to (x - 1) do {
       numbers[i] = read()
     }
     call heapSort(x) ;
     for i = 0 to (x - 1) do {
       write(numbers[i])
     }
     return 0
```

Figure 10.1: Heap-sort in SIMPLE

10.1 Untyped SIMPLE

K-Annotated Syntax of Untyped SIMPLE

$$
\begin{array}{rcll}
\mathit{VarId} & ::= & \text{all identifiers; to be used as names of variables and functions} & \\
\mathit{Exp} & ::= & \mathit{Int} \mid \mathit{Bool} \mid \mathit{VarId} & \\
& \mid & \mathit{VarId}[\mathit{Exp}] & [\mathit{strict}(2)] \\
& \mid & \mathsf{read}() & \\
& \mid & \mathit{VarId}\ \mathbf{List}^{()}_{\_,\_}\{\mathit{Exp}\} & [\mathit{strict}] \\
& \mid & \mathit{Exp} + \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ +_{\mathit{Int}\times\mathit{Int}\to\mathit{Int}}] \\
& \mid & \mathit{Exp} - \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ -_{\mathit{Int}\times\mathit{Int}\to\mathit{Int}}] \\
& \mid & \mathit{Exp} * \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ *_{\mathit{Int}\times\mathit{Int}\to\mathit{Int}}] \\
& \mid & \mathit{Exp} / \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \mathit{quotient}_{\mathit{Int}\times\mathit{Int}\to\mathit{Int}}] \\
& \mid & \mathit{Exp} \% \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \mathit{remainder}_{\mathit{Int}\times\mathit{Int}\to\mathit{Int}}] \\
& \mid & -\mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ -_{\mathit{Int}\to\mathit{Int}}] \\
& \mid & \mathit{Exp} < \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ <_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp} <= \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \leq_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp} > \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ >_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp} >= \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \geq_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp} == \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ =_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp} != \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \neq_{\mathit{Int}\times\mathit{Int}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp}\ \mathsf{and}\ \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \wedge_{\mathit{Bool}\times\mathit{Bool}\to\mathit{Bool}}] \\
& \mid & \mathit{Exp}\ \mathsf{or}\ \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \vee_{\mathit{Bool}\times\mathit{Bool}\to\mathit{Bool}}] \\
& \mid & \mathsf{not}\ \mathit{Exp} & [\mathit{strict},\ \mathit{extends}\ \neg_{\mathit{Bool}\to\mathit{Bool}}] \\
\mathit{Decl} & ::= & \mathsf{var}\ \mathit{Name} \mid \mathsf{var}\ \mathit{Name}[\mathit{Nat}] & \\
& \mid & \mathit{Decl}; \mathit{Decl} & [d_1; d_2 \rightharpoonup d_1 \curvearrowright d_2] \\
\mathit{Stmt} & ::= & \{\} & [\{\} \rightharpoonup \cdot] \\
& \mid & \{\mathit{Stmt}\} & [\{s\} \rightharpoonup s] \\
& \mid & \{\mathit{Decl}; \mathit{Stmt}\} & \\
& \mid & \mathit{Stmt}; \mathit{Stmt} & [s_1; s_2 \rightharpoonup s_1 \curvearrowright s_2] \\
& \mid & \mathsf{write}(\mathit{Exp}) & [\mathit{strict}] \\
& \mid & \mathit{VarId} = \mathit{Exp} & [\mathit{strict}(2)] \\
& \mid & \mathit{VarId}[\mathit{Exp}] = \mathit{Exp} & [\mathit{strict}(2,3)] \\
& \mid & \mathsf{if}\ \mathit{Exp}\ \mathsf{then}\ \mathit{Stmt}\ \mathsf{else}\ \mathit{Stmt} & [\mathit{strict}(1)] \\
& \mid & \mathsf{if}\ \mathit{Exp}\ \mathsf{then}\ \mathit{Stmt} & [\mathsf{if}\ e\ \mathsf{then}\ s \rightharpoonup \mathsf{if}\ e\ \mathsf{then}\ s\ \mathsf{else}\ \cdot] \\
& \mid & \mathsf{while}\ \mathit{Exp}\ \mathsf{do}\ \mathit{Stmt} & \\
& \mid & \mathsf{for}\ \mathit{VarId} = \mathit{Exp}\ \mathsf{to}\ \mathit{Exp}\ \mathsf{do}\ \mathit{Stmt} & \\
& & \quad [\mathsf{for}\ i = e_1\ \mathsf{to}\ e_2\ \mathsf{do}\ s \rightharpoonup \{\mathsf{var}\ i;\ i = e_1;\ \mathsf{while}\ (i <= e_2)\ \mathsf{do}\ \{s; i = i + 1\}\}] & \\
& \mid & \mathsf{call}\ \mathit{Exp} & [\mathit{strict}] \\
& \mid & \mathsf{return}\ \mathit{Exp} & [\mathit{strict}] \\
\mathit{FunDecl} & ::= & \mathsf{function}\ \mathit{VarId}\ \mathbf{List}^{()}_{\_,\_}\{\mathit{VarId}\}\ \mathit{Stmt} & \\
& \mid & \mathbf{List}_{\_\_\_}\{\mathit{FunDecl}\} & \\
\mathit{Pgm} & ::= & \mathit{FunDecl} & \\
& \mid & \mathit{Decl}; \mathit{FunDecl} & [d; fd \rightharpoonup d \curvearrowright fd]
\end{array}
$$

K Semantics of Untyped SILF

$\langle \underset{\mathsf{call\ main()}}{\underline{fd}} \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle \underset{\rho}{\underline{\cdot}} \rangle_{\mathsf{genv}}\ \langle \underset{fd}{\underline{\cdot}} \rangle_{\mathsf{functions}}$

$\langle \underset{v}{\underline{x}}\ \cdots \rangle_{\mathsf{k}}\ \langle \cdots\ x \mapsto l\ \cdots \rangle_{\mathsf{env}}\ \langle \cdots\ l \mapsto v\ \cdots \rangle_{\mathsf{store}}$

$\langle \underset{\sigma[\rho[x] +_{Loc} n]}{\underline{x[n]}}\ \cdots \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle \sigma \rangle_{\mathsf{store}}$

$\langle \underset{i}{\underline{\mathsf{read()}}}\ \cdots \rangle_{\mathsf{k}}\ \langle \underset{\cdot}{\underline{i}}\ \cdots \rangle_{in}$

$\langle \underset{s}{\underline{f(vl) \curvearrowright k}} \rangle_{\mathsf{k}}\ \langle \underset{(\rho, k)}{\underline{\cdot}}\ \cdots \rangle_{fstack}\ \langle \underset{\rho'[ll/xl]}{\underline{\rho}} \rangle_{\mathsf{env}}\ \langle \rho' \rangle_{\mathsf{genv}}\ \langle \underset{\sigma[vl/ll]}{\underline{\sigma}} \rangle_{\mathsf{store}}\ \langle \underset{l +_{Int} |xl|}{\underline{l}} \rangle_{\mathsf{nextLoc}}\ \langle \cdots\ \mathsf{function}\ f(xl)\ s\ \cdots \rangle_{\mathsf{functions}}$ wh
l

$\langle \underset{\cdot}{\underline{\mathsf{var}\ x}}\ \cdots \rangle_{\mathsf{k}}\ \langle \underset{\rho[l/x]}{\underline{\rho}} \rangle_{\mathsf{env}}\ \langle \underset{l +_{Loc} 1}{\underline{l}} \rangle_{\mathsf{nextLoc}}$

$\langle \underset{\cdot}{\underline{\mathsf{var}\ x[n]}}\ \cdots \rangle_{\mathsf{k}}\ \langle \underset{\rho[l/x]}{\underline{\rho}} \rangle_{\mathsf{env}}\ \langle \underset{l +_{Loc} n}{\underline{l}} \rangle_{\mathsf{nextLoc}}$

$\langle \underset{d \curvearrowright s \curvearrowright \mathsf{env}(\rho)}{\underline{\{d; s\}}}\ \cdots \rangle_{\mathsf{k}}\ \mathsf{env}(\rho)$

$\langle \underset{\cdot}{\underline{\mathsf{env}(\rho)}}\ \cdots \rangle_{\mathsf{k}}\ \langle \underset{\rho}{\underline{\_}} \rangle_{\mathsf{env}}$

$\langle \underset{\cdot}{\underline{\mathsf{write}\ i}}\ \cdots \rangle_{\mathsf{k}}\ \langle \cdots\ \underset{i}{\underline{\cdot}} \rangle_{out}$

$\langle \underset{\cdot}{\underline{x = v}}\ \cdots \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle \underset{\sigma[v/\rho[x]]}{\underline{\sigma}} \rangle_{\mathsf{store}}$

$\langle \underset{\cdot}{\underline{x[n] = v}}\ \cdots \rangle_{\mathsf{k}}\ \langle \rho \rangle_{\mathsf{env}}\ \langle \underset{\sigma[v/\rho[x] +_{Loc} n]}{\underline{\sigma}} \rangle_{\mathsf{store}}$

$\mathsf{if\ true\ then}\ s_1\ \mathsf{else}\ s_2 \to s_1$
$\mathsf{if\ false\ then}\ s_1\ \mathsf{else}\ s_2 \to s_2$
$\langle \mathsf{while}\ b\ \mathsf{do}\ s\ \cdots \rangle_{\mathsf{k}} \rightharpoonup \langle \mathsf{if}\ b\ \mathsf{then}\ (s; \mathsf{while}\ b\ \mathsf{do}\ s)\ \cdots \rangle_{\mathsf{k}}$
$\mathsf{call}\ v \to .$
$\langle \underset{v \curvearrowright k}{\underline{\mathsf{return}\ v \curvearrowright \_}} \rangle_{\mathsf{k}}\ \langle \underset{\cdot}{\underline{(\rho, k)}}\ \cdots \rangle_{fstack}\ \langle \underset{\rho}{\underline{\_}} \rangle_{\mathsf{env}}$