```
void sort( int a[], int low, int high)
{ int i; int k;
  i = low;
  while (i < high-1)
    { int t;
      k = minloc(a,i,high);
      t = a[k];
      a[k] = a[i];
      a[i] = t;
      i = i + 1;
    }
}

void main(void)
{ int i;
  i = 0;
  while (i < 10)
    { x[i] = input();
      i = i + 1; }
  sort(x,0,10);
  i = 0;
  while (i < 10)
    { output(x[i]);
      i = i + 1; }
}
```

A.4 A TINY MACHINE RUNTIME ENVIRONMENT FOR THE C– LANGUAGE

The following description assumes a knowledge of the Tiny Machine as given in Section 8.7 and an understanding of stack-based runtime environments from Chapter 7. Since C– (unlike TINY) has recursive procedures, the runtime environment must be stack based. The environment consists of a global area at the top of dMem, and the stack just below it, growing downward toward 0. Since C– contains no pointers or dynamic allocation, there is no need for a heap. The basic organization of each activation record (or stack frame) in C– is