```
template <typename T>
void heapInsert(fstream& f, int k, int n, T& dummy)
{ int i, j = 2*k + 1;
  T a, b;
  while (j < n)
    { i = j;
      f.seekg(j*sizeof(T));
      f.read((char*)&a, sizeof(T));
      if (j < n-1)
        { f.read((char*)&b, sizeof(T));
          if (a < b) i = j+1;
        }
      f.seekg(i*sizeof(T));
      f.read((char*)&b, sizeof(T));
      f.seekg(k*sizeof(T));
      f.read((char*)&a, sizeof(T));
      if (a < b)
        { swap(f, k, i, dummy);
          k = i;
          j = 2*k+1;
        }
      else j = n;
    }
}

template <typename T>
void heapSort(fstream& f, T dummy)
{ f.seekg(0, ios::end);
  int n = f.tellg()/sizeof(T);
  int i = n/2-1;
  for (int j = i; j >= 0; j--)
     heapInsert(f, j, n, dummy);
  n--;
  while (n > 0)
    { swap(f, 0, n, dummy);
      heapInsert(f, 0, n, dummy);
      n--;
    }
  f.close();
}
```