

```

from cctbx.array_family import flex
from cctbx import crystal
from cctbx import uctbx
from cctbx import sgtbx
from cctbx import miller
import sys

def run(args):
    assert len(args) == 1
    lines = open(args[0]).read().splitlines()
    title = lines[0]
    unit_cell = uctbx.unit_cell(lines[1])
    n_symops = int(lines[2].split()[0])
    space_group = sgtbx.space_group()
    for line in lines[3:3+n_symops]:
        coeffs = [float(field) for field in line.split()]
        space_group.expand_smx(sgtbx.rt_mx(coeffs[:9], coeffs[9:]))
    crystal_symmetry = crystal.symmetry(
        unit_cell=unit_cell,
        space_group=space_group)
    miller_indices = flex.miller_index()
    data = flex.double()
    sigmas = flex.double()
    for i_line in xrange(3+n_symops, len(lines)):
        fields = lines[i_line].split()
        assert len(fields) == 5
        miller_indices.append([int(value) for value in fields[:3]])
        data.append(float(fields[3]))
        sigmas.append(float(fields[4]))
    miller_set=miller.set(
        crystal_symmetry=crystal_symmetry,
        indices=miller_indices,
        anomalous_flag=False)
    miller_array = miller_set.array(
        data=data,
        sigmas=sigmas).set_observation_type_xray_intensity()
    print "Before merging:"
    miller_array.show_summary()
    print
    merged = miller_array.merge_equivalents().array().sort(by_value="data")
    print "After merging:"
    merged.show_comprehensive_summary().show_array()
    print

if (__name__ == "__main__"):
    run(sys.argv[1:])

```