/\*

\* mergesort.cpp

\* Implementation of a bitonic mergesort

\*/

/\* merge(input, size, output, asc)

Merge the two halves of the array input (which has size elements) into

output. If asc is true, then the output array should be in ascending order;

otherwise it should be descending.

\*/

void merge(int\* input, int size, int\* output, bool output\_asc) {

// Your merge implementation goes here

}

/\* mergesort(input, size, output, asc)

Mergesort the input array (with size elements) into the output array. If

asc is true, the output array should be sorted ascending, otherwise it should

be descending.

\*/

void mergesort(int \*input, int size, int\* output, bool output\_asc) {

// Your mergesort implementation goes here

}

/\* mergesort(input, size)

Sorts size elements in the array pointed to by input, using the MergeSort

algorithm. Output is returned as a newly allocated array, which the caller

is responsible for freeing.

\*/

int\* mergesort(int\* input, int size) {

int\* output = new int[size];

mergesort(input, size, output, true);

return output;

}