Implement Min and Max operations using Parallel Reduction.

```
#include <iostream>
#include <vector>
#include <omp.h>

int parallelMin(const std::vector<int>& arr) {
   int minVal = arr[0];

   #pragma omp parallel for reduction(min: minVal)
   for (int i = 1; i < arr.size(); ++i) {
      if (arr[i] < minVal) {
         minVal = arr[i];
      }
   }

   return minVal;
}

int parallelMax(const std::vector<int>& arr) {
   int maxVal = arr[0];

   #pragma omp parallel for reduction(max: maxVal)
   for (int i = 1; i < arr.size(); ++i) {
      if (arr[i] > maxVal) {
         maxVal = arr[i];
      }
   }

   return maxVal;
}

int main() {
   std::vector<int> arr = {9, 4, 2, 7, 5, 1, 8, 3, 6};

   int minVal = parallelMin(arr);
   int maxVal = parallelMax(arr);

   std::cout << "Minimum value: " << minVal << std::endl;
   std::cout << "Maximum value: " << maxVal << std::endl;

   return 0;
}
```