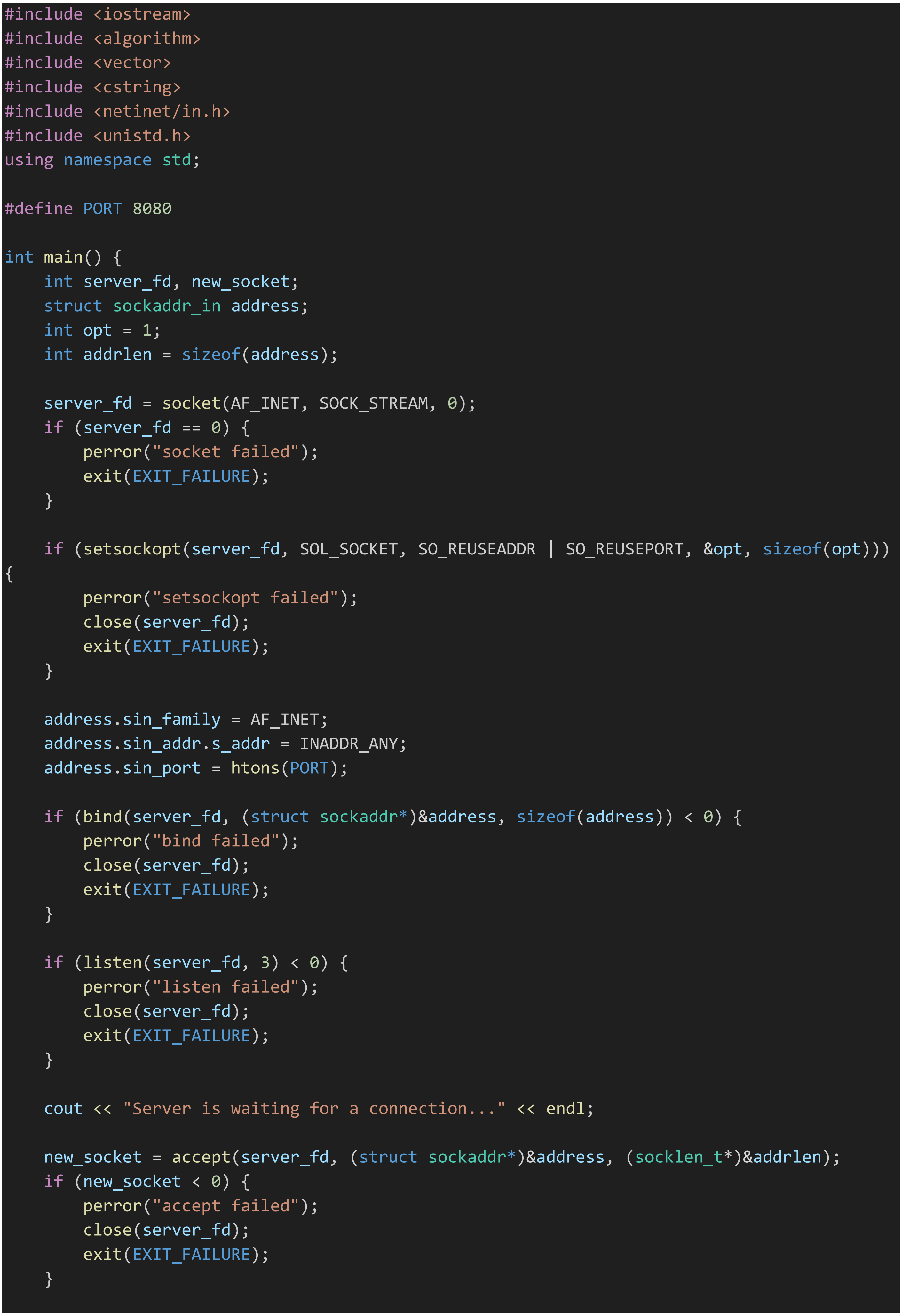

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <cstring>
#include <netinet/in.h>
#include <unistd.h>
using namespace std;

#define PORT 8080

int main() {
    int server_fd, new_socket;
    struct sockaddr_in address;
    int opt = 1;
    int addrlen = sizeof(address);

    server_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (server_fd == 0) {
        perror("socket failed");
        exit(EXIT_FAILURE);
    }

    if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR | SO_REUSEPORT, &opt, sizeof(opt)))
{
        perror("setsockopt failed");
        close(server_fd);
        exit(EXIT_FAILURE);
    }

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    if (bind(server_fd, (struct sockaddr*)&address, sizeof(address)) < 0) {
        perror("bind failed");
        close(server_fd);
        exit(EXIT_FAILURE);
    }

    if (listen(server_fd, 3) < 0) {
        perror("listen failed");
        close(server_fd);
        exit(EXIT_FAILURE);
    }

    cout << "Server is waiting for a connection..." << endl;

    new_socket = accept(server_fd, (struct sockaddr*)&address, (socklen_t*)&addrlen);
    if (new_socket < 0) {
        perror("accept failed");
        close(server_fd);
        exit(EXIT_FAILURE);
    }
    cout << "Connection established with client." << endl;

    int arr_size;
    read(new_socket, &arr_size, sizeof(arr_size));

    vector<int> numbers(arr_size);
    read(new_socket, numbers.data(), arr_size * sizeof(int));

    cout << "Received array: ";
    for (int num : numbers) cout << num << " ";
    cout << endl;

    sort(numbers.begin(), numbers.end());

    send(new_socket, numbers.data(), arr_size * sizeof(int), 0);

    cout << "Sorted array sent back to client." << endl;

    close(new_socket);
    close(server_fd);
    return 0;
}
```