**Code to handle the signals**

#include <iostream>

#include <csignal>

#include <unistd.h>

#include <cstring>

// Signal handler functions

void handle\_signal(int signum) {

std::cout << "Received signal: " << signum << " (" << strsignal(signum) << ")\n";

}

void mask\_signal(int signum) {

std::cout << "Signal: " << signum << " (" << strsignal(signum) << ") is masked.\n";

}

int main() {

struct sigaction sa;

sigset\_t mask;

// Set up the signal handler for signals 1 to 9

sa.sa\_handler = handle\_signal;

sa.sa\_flags = 0;

sigemptyset(&sa.sa\_mask);

for (int i = 1; i <= 9; ++i) {

if (i != 2 && i != 3) { // Exclude SIGINT (2) and SIGQUIT (3) from handling

if (sigaction(i, &sa, nullptr) == -1) {

std::cerr << "Error setting handler for signal " << i << ": " << strerror(errno) << "\n";

}

}

}

// Mask signals SIGINT (2) and SIGQUIT (3)

sigemptyset(&mask);

sigaddset(&mask, SIGINT);

sigaddset(&mask, SIGQUIT);

if (sigprocmask(SIG\_BLOCK, &mask, nullptr) == -1) {

std::cerr << "Error blocking signals: " << strerror(errno) << "\n";

return 1;

}

std::cout << "Program started. PID: " << getpid() << "\n";

// Infinite loop to keep the program running

while (true) {

std::cout << "Running...\n";

sleep(1); // Sleep for 1 second

}

return 0;

}