Sample Text from Roboto Mono Regular

```
#include <iostream>
#include <algorithm>
#include <functional>
#include <iterator>
#include <cstdlib>
#include <ctime>

#define BIT_MASK 0xDEADBEAF

#define MULTILINE_DEF(a,b) if ((a)>2) { \
auto temp = (b)/2; \
(b)+=10; \
someFunctionCall((a),(b));\
}

namespace LevelOneNamespace {
namespace LevelTwoNamespace {

template <typename T, int size> bool is_sorted(T(&array)[size]) {
  return std::adjacent_find(array, array + size, std::greater<T>()) ==
         array + size;
}

std::vector<uint32_t> returnVector( uint32_t* LongNameForParameter1,
                                    double* LongNameForParameter2,
                                    const float& LongNameForParameter3,
                                    const
std::map<std::string,int32_t>& LongNameForParameter4) {

    //TODO: This is a long comment that allows you to understand how
long comments will be trimmed. Here should be deep thought but it's
just not right time for this

    for (auto& i: LongNameForParameter4) {
        auto b =
someFunctionCall(static_cast<int16_t>(*LongNameForParameter2),reinterpret_cast
        i.second++;
    }

    do {
    if (a)
        a--;
    else
        a++;
    } while (false);

    return {};
}

}
}

int main() {
  std::srand(std::time(0));

  int list[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

  do {
    std::random_shuffle(list, list + 9);
```

Sample Text from Roboto Mono Regular

```
  } while (is_sorted(list));

  int score = 0;

  do {
    std::cout << "Current list: ";
    std::copy(list, list + 9, std::ostream_iterator<int>(std::cout, "
"));

    int rev;
    while (true) {
      std::cout << "\nDigits to reverse? ";
      std::cin >> rev;
      if (rev > 1 && rev < 10)
        break;
      std::cout << "Please enter a value between 2 and 9.";
    }

    ++score;
    std::reverse(list, list + rev);
  } while (!is_sorted(list));

  std::cout << "Congratulations, you sorted the list.\n"
            << "You needed " << score << " reversals." << std::endl;
  return 0;
}
```