

Выполнение работы

Задание 1: «Генератор ежедневных температур»

Задача: сгенерировать массив температур для небольшого городка за 30 дней.

The screenshot shows a terminal window with the following content:

```
src > main.cpp > ...
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main() {
6     double data[30];
7     for (int i = 0; i < 30; i++) {
8         if (i == 0) {
9             data[i] = ((rand() % 450) - 150) / 10.0;
10        }
11
12        data[i] = data[i - 1] + ((rand() % 60) - 30) / 10.0;
13
14        if (data[i] < -15.0) {
15            data[i] += 6.0;
16        }
17
18        if (data[i] > 30.0) {
19            data[i] -= 6.0;
20        }
21
22        std::cout << "День " << i << ": " << data[i] << "\n";
23    }
24 }
```

Below the code, the terminal output shows the generated temperatures for each day from 0 to 29:

```
[gregory@HP testing]$ cd ./build
[gregory@HP build]$ ./app
День 0: 1.6
День 1: 4.3
День 2: 6.8
День 3: 9.1
День 4: 11.6
День 5: 13.2
День 6: 11.4
День 7: 9.3
День 8: 6.4
День 9: 3.6
День 10: 1.3
День 11: 3.3
День 12: 2.2
День 13: 1.5
День 14: 3.1
День 15: 0.1
День 16: -2.3
День 17: -0.1
День 18: -1.5
День 19: -3.4
День 20: -5.6
День 21: -5.9
День 22: -8
День 23: -10.8
День 24: -8.8
День 25: -11.6
День 26: -10.3
День 27: -12.6
День 28: -10.1
День 29: -10.2
```

Рисунок 1 – выполнение задания 1

Задача 2: «Генератор оценок студентов»

Задача: создать журнал успеваемости группы из 15 студентов по 5 предметам.

```
src > ➜ main.cpp > clangd > main
 1 #include <cstdlib>
 2 #include <ctime>
 3 #include <iostream>
 4
 5 int main() {
 6     int grades[15][5];
 7     int a = (rand() % 7);
 8     int b = a + 5;
 9
10     for (int i = 0; i < 15; i++) {
11         for (int j = 0; j < 5; j++) {
12             if ((i == a) || (i == b)) {
13                 grades[i][j] = (rand() % 2) + 4;
14             } else {
15                 grades[i][j] = (rand() % 4) + 2;
16             }
17         }
18     }
19
20     for (int i = 0; i < 15; i++) {
21         std::cout << "Студент " << i + 1 << ": ";
22         for (int j = 0; j < 5; j++) {
23             std::cout << grades[i][j] << " ";
24         }
25         std::cout << "\n";
26     }
27
28     return 0;
29 }
30
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[ 50%] Building CXX object CMakeFiles/app.dir/src/main.cpp.o
[100%] Linking CXX executable app
[100%] Built target app
Студент 1: 4 3 5 3 5
Студент 2: 4 4 5 5 4
Студент 3: 5 4 5 5 4
Студент 4: 2 4 2 2 5
Студент 5: 2 5 3 4 4
Студент 6: 4 5 5 5 3
Студент 7: 4 4 4 5 5
Студент 8: 3 2 5 4 3
Студент 9: 3 3 5 2 3
Студент 10: 4 2 5 4 3
Студент 11: 4 5 2 2 3
Студент 12: 4 4 2 3 3
Студент 13: 3 2 5 2 3
Студент 14: 4 3 3 3 2
Студент 15: 5 4 3 4 5
[gregory@HP build]$
```

Рисунок 2 – выполнение задания 2

Задание 3: «Генератор ежечасных продаж кофе»

Задача: сгенерировать данные о продажах кофе в кофейне за неделю по часам.

```
src > clangd > main
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main() {
6     int coffe_sales[7][14];
7
8     for (int i = 0; i < 7; i++) {
9         for (int j = 0; j < 14; j++) {
10
11             if ((i != 5) && (i != 6)) {
12                 switch (j) {
13                     case 3: coffe_sales[i][j] = (rand() % 20) + 20; break; // 11:00, Утренние часы (8-11) 20-40 чашек/ч
14                     case 6: coffe_sales[i][j] = (rand() % 30) + 30; break; // 14:00, Дневные часы (12-14) 30-60 чашек/ч
15                     case 9: coffe_sales[i][j] = (rand() % 10) + 5; break; // 17:00, Остальные часы 5-15 чашек/ч
16                     case 12: coffe_sales[i][j] = (rand() % 20) + 15; break; // 20:00, Вечерние часы (17-20) 15-35 чашек/ч
17                     default: coffe_sales[i][j] = (rand() % 10) + 5; break; // Остальные часы
18                 }
19             } else {
20                 switch (j) {
21                     case 3: coffe_sales[i][j] = ((rand() % 20) + 20) * 1.2; break; // 11:00, Утренние часы (8-11)
22                     case 6: coffe_sales[i][j] = ((rand() % 30) + 30) * 1.2; break; // 14:00, Дневные часы (12-14)
23                     case 9: coffe_sales[i][j] = ((rand() % 10) + 5) * 1.2; break; // Остальные часы (5-15)
24                     case 12: coffe_sales[i][j] = ((rand() % 20) + 15) * 1.2; break; // 20:00, Вечерние часы (17-20)
25                     default: coffe_sales[i][j] = ((rand() % 10) + 5) * 1.2; break; // Остальные часы
26                 }
27             }
28         }
29     }
30
31     for (int i = 0; i < 7; i++) {
32         for (int j = 3; j < 14; j++) {
33             switch (i) {
34                 case 0: std::cout << "Понедельник, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
35                 case 1: std::cout << "Вторник, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
36                 case 2: std::cout << "Среда, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
37                 case 3: std::cout << "Четверг, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
38                 case 4: std::cout << "Пятница, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
39                 case 5: std::cout << "Суббота, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
40                 case 6: std::cout << "Воскресенье, " << j + 8 << ":00 = " << coffe_sales[i][j] << " чашек\n"; break;
41             }
42         }
43     }
44
45     return 0;
46 }
```

Рисунок 3 – выполнение задания 3

```
[100%] Built target app
Понедельник, 11:00 = 35 чашек
Понедельник, 12:00 = 8 чашек
Понедельник, 13:00 = 10 чашек
Понедельник, 14:00 = 46 чашек
Понедельник, 15:00 = 7 чашек
Понедельник, 16:00 = 14 чашек
Понедельник, 17:00 = 6 чашек
Понедельник, 18:00 = 7 чашек
Понедельник, 19:00 = 12 чашек
Понедельник, 20:00 = 25 чашек
Понедельник, 21:00 = 14 чашек
Вторник, 11:00 = 26 чашек
Вторник, 12:00 = 7 чашек
Вторник, 13:00 = 11 чашек
Вторник, 14:00 = 41 чашек
Вторник, 15:00 = 13 чашек
Вторник, 16:00 = 12 чашек
Вторник, 17:00 = 14 чашек
Вторник, 18:00 = 7 чашек
Вторник, 19:00 = 5 чашек
Вторник, 20:00 = 17 чашек
Вторник, 21:00 = 8 чашек
Среда, 11:00 = 22 чашек
Среда, 12:00 = 7 чашек
Среда, 13:00 = 13 чашек
Среда, 14:00 = 59 чашек
Среда, 15:00 = 12 чашек
Среда, 16:00 = 8 чашек
Среда, 17:00 = 11 чашек
Среда, 18:00 = 6 чашек
Среда, 19:00 = 7 чашек
Среда, 20:00 = 24 чашек
Среда, 21:00 = 8 чашек
Четверг, 11:00 = 37 чашек
Четверг, 12:00 = 13 чашек
Четверг, 13:00 = 9 чашек
Четверг, 14:00 = 35 чашек
Четверг, 15:00 = 5 чашек
Четверг, 16:00 = 8 чашек
Четверг, 17:00 = 11 чашек
Четверг, 18:00 = 6 чашек
Четверг, 19:00 = 5 чашек
Четверг, 20:00 = 31 чашек
Четверг, 21:00 = 8 чашек
Пятница, 11:00 = 21 чашек
Пятница, 12:00 = 10 чашек
Пятница, 13:00 = 10 чашек
Пятница, 14:00 = 44 чашек
Пятница, 15:00 = 12 чашек
Пятница, 16:00 = 11 чашек
Пятница, 17:00 = 10 чашек
Пятница, 18:00 = 11 чашек
Пятница, 19:00 = 14 чашек
Пятница, 20:00 = 28 чашек
Пятница, 21:00 = 12 чашек
Суббота, 11:00 = 30 чашек
Суббота, 12:00 = 10 чашек
Суббота, 13:00 = 14 чашек
Суббота, 14:00 = 52 чашек
Суббота, 15:00 = 10 чашек
Суббота, 16:00 = 9 чашек
Суббота, 17:00 = 6 чашек
Суббота, 18:00 = 14 чашек
Суббота, 19:00 = 15 чашек
Суббота, 20:00 = 37 чашек
Суббота, 21:00 = 15 чашек
Воскресенье, 11:00 = 37 чашек
Воскресенье, 12:00 = 10 чашек
Воскресенье, 13:00 = 16 чашек
Воскресенье, 14:00 = 50 чашек
Воскресенье, 15:00 = 6 чашек
Воскресенье, 16:00 = 13 чашек
Воскресенье, 17:00 = 15 чашек
Воскресенье, 18:00 = 16 чашек
Воскресенье, 19:00 = 8 чашек
Воскресенье, 20:00 = 25 чашек
Воскресенье, 21:00 = 13 чашек
```

Рисунок 4 – вывод задания 3

Задание 4: «Генератор случайных паролей»

Задача: сгенерировать 10 случайных паролей заданной длины.

```
src > G main.cpp > clangd > ⚡ main
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 int main() {
6     char passwords[10][9];
7
8     for (int i = 0; i < 10; i++) {
9         passwords[i][0] = (rand() % 26) + 65; // первая буква – заглавная
10        for (int j = 1; j < 9; j++) {
11            int a = rand() % 3;
12            switch (a) {
13                case 0: passwords[i][j] = (rand() % 26) + 65; break;
14                case 1: passwords[i][j] = (rand() % 26) + 97; break;
15                case 2: passwords[i][j] = (rand() % 10) + 48; break;
16            }
17        }
18    }
19
20    for (int i = 0; i < 10; i++) {
21        std::cout << "Пароль " << i + 1 << ":" ;
22        for (int j = 0; j < 9; j++) {
23            std::cout << passwords[i][j];
24        }
25        std::cout << "\n";
26    }
27 }
```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[gregory@HP testing]$ rm -rf build && cmake -B build && cmake --build build && cd ./build && ./app && cd ..
[100%] Built target app
Пароль 1: NlbmBcazw
Пароль 2: Ky8D03r2W
Пароль 3: FXjbD4s5B
Пароль 4: Ye30xP751
Пароль 5: L4AQW37M8
Пароль 6: QN9U8smBU
Пароль 7: QlJ7W16b4
Пароль 8: XvRBTS3Z
Пароль 9: Q98A3000C
Пароль 10: N51f88wq4
```

Рисунок 5 – выполнение задания 4

Задание 5: «Генератор графика дел на день»

Задача: создать расписание дел на день с временными интервалами.

```
src > G main.cpp > ...
1 #include <iostream>
2 #include <vector>
3 #include <ctime>
4 #include <cstdlib>
5 #include <algorithm>
6 #include <iomanip>
7
8 struct Task {
9     int start_hour;
10    int start_minute;
11    int duration;
12 };
13
14 bool areTasksOverlapping(const Task& t1, const Task& t2) {
15     int t1_start_total = t1.start_hour * 60 + t1.start_minute;
16     int t1_end_total = t1_start_total + t1.duration;
17     int t2_start_total = t2.start_hour * 60 + t2.start_minute;
18     int t2_end_total = t2_start_total + t2.duration;
19     return !(t1_end_total <= t2_start_total || t2_end_total <= t1_start_total);
20 }
21
22 Task generateRandomTask(int min_start_hour, int max_end_hour_exclusive) {
23     Task newTask{};
24     bool validTask = false;
25     while (!validTask) {
26         newTask.start_hour = min_start_hour + rand() % (max_end_hour_exclusive - min_start_hour);
27         newTask.start_minute = rand() % 60;
28         newTask.duration = 15 + rand() % (120 - 15 + 1);
29         int end_total = newTask.start_hour * 60 + newTask.start_minute + newTask.duration;
30         if (end_total <= max_end_hour_exclusive * 60) validTask = true;
31     }
32     return newTask;
33 }
34
35 int main() {
36     std::srand(static_cast<unsigned>(std::time(nullptr)));
37
38     std::vector<Task> schedule;
39     const int numTasks = 10;
40
41     while (schedule.size() < static_cast<size_t>(numTasks)) {
42         Task newTask = generateRandomTask(min_start_hour: 8, max_end_hour_exclusive: 22);
43         bool overlaps = false;
44         for (const auto& existing : schedule) {
45             if (areTasksOverlapping(newTask, existing)) { overlaps = true; break; }
46         }
47         if (!overlaps) {
48             auto it = std::lower_bound(
49                 schedule.begin(), schedule.end(), newTask,
50                 [] (const Task& a, const Task& b) {
51                     if (a.start_hour != b.start_hour) return a.start_hour < b.start_hour;
52                     return a.start_minute < b.start_minute;
53                 });
54             schedule.insert(it, newTask);
55         }
56     }
57
58     std::cout << "Расписание на день:\n";
59     for (size_t i = 0; i < schedule.size(); ++i) {
60         const Task& t = schedule[i];
61         int end_hour = (t.start_hour * 60 + t.start_minute + t.duration) / 60;
62         int end_minute = (t.start_minute + t.duration) % 60;
63
64         std::cout
65             << std::setw(2) << std::setfill('0') << t.start_hour << ':'
66             << std::setw(2) << std::setfill('0') << t.start_minute << " - "
67             << std::setw(2) << std::setfill('0') << end_hour << ':'
68             << std::setw(2) << std::setfill('0') << end_minute
69             << " Дело " << (i + 1) << '\n';
70     }
71 }
72
73 return 0;
74 }
75
```

PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS
[gregory@IP testing]\$ rm -rf build && cmake -B build && cmake --build build && cd ./build && ./app && cd ..
[100%] Built target app
Расписание на день:
08:10 - 09:05 Дело 1
09:05 - 09:23 Дело 2
10:42 - 11:22 Дело 3
11:48 - 11:55 Дело 4
12:22 - 14:15 Дело 5
15:08 - 16:39 Дело 6
17:14 - 17:56 Дело 7
18:30 - 19:11 Дело 8
19:19 - 19:36 Дело 9
19:41 - 20:18 Дело 10

Рисунок 6 – выполнение задания 5

Задание 6: «Генератор цветовой палитры»

Задача: сгенерировать 5 гармоничных цветовых палитр в RGB формате.

```
src > main.cpp > ...
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4
5  struct color {
6      int red;
7      int green;
8      int blue;
9  };
10
11 int main() {
12     std::srand(static_cast<unsigned>(std::time(nullptr)));
13
14     color palettes[5][4];
15
16     for (int i = 0; i < 5; i++) {
17         // базовый цвет
18         palettes[i][0].red = std::rand() % 256;
19         palettes[i][0].green = std::rand() % 256;
20         palettes[i][0].blue = std::rand() % 256;
21
22         // оттенки/контраст
23         for (int j = 1; j < 4; j++) {
24             switch (j) {
25                 case 1:
26                     palettes[i][j].red = palettes[i][0].red + (std::rand() % 40) - 20;
27                     palettes[i][j].green = palettes[i][0].green + (std::rand() % 40) - 20;
28                     palettes[i][j].blue = palettes[i][0].blue + (std::rand() % 40) - 20;
29                     break;
30                 case 2:
31                     palettes[i][j].red = palettes[i][0].red + (std::rand() % 40) - 20;
32                     palettes[i][j].green = palettes[i][0].green + (std::rand() % 40) - 20;
33                     palettes[i][j].blue = palettes[i][0].blue + (std::rand() % 40) - 20;
34                     break;
35                 case 3:
36                     palettes[i][j].red = 255 - palettes[i][0].red;
37                     palettes[i][j].green = 255 - palettes[i][0].green;
38                     palettes[i][j].blue = 255 - palettes[i][0].blue;
39                     break;
40             }
41         }
42     }
43
44     for (int i = 0; i < 5; i++) {
45         std::cout << "Палитра " << (i + 1) << ":" \n";
46         for (int j = 0; j < 4; j++) {
47             std::cout << "Цвет " << (j + 1) << ": (" <<
48             << palettes[i][j].red << ", "
49             << palettes[i][j].green << ", "
50             << palettes[i][j].blue << ")" \n;
51         }
52         std::cout << "\n";
53     }
54
55     return 0;
56 }
57
```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[gregory@HP testing]$ rm -rf build && cmake -B build && cmake --build build && cd ./build && ./app && cd ..
Палитра 1:
Цвет 1: (152, 227, 119)
Цвет 2: (132, 236, 125)
Цвет 3: (156, 219, 136)
Цвет 4: (103, 28, 136)

Палитра 2:
Цвет 1: (220, 219, 2)
Цвет 2: (206, 232, -5)
Цвет 3: (227, 200, -6)
Цвет 4: (35, 36, 253)

Палитра 3:
Цвет 1: (251, 200, 44)
Цвет 2: (242, 184, 44)
Цвет 3: (244, 211, 57)
Цвет 4: (4, 55, 211)

Палитра 4:
Цвет 1: (13, 110, 251)
Цвет 2: (4, 104, 237)
Цвет 3: (3, 97, 258)
Цвет 4: (242, 145, 4)

Палитра 5:
Цвет 1: (108, 95, 79)
Цвет 2: (113, 110, 61)
Цвет 3: (115, 108, 95)
Цвет 4: (147, 160, 176)
```

Рисунок 7 – выполнение задания 6