

```
1  #include <stdlib.h>
2  #include <iostream>
3
4  using namespace std;
5
6  /*
7   * Вывод на экран десяти членов арифметической прогрессии
8   * случай 1
9   */
10 int main()
11 {
12     //Задаем параметры прогрессии
13     int a1 = 4;
14     int d = 3;
15
16     //задаем переменную для хранения i-го члена прогрессии
17     int ai;
18
19     //Количество элементов
20     int n = 10;
21
22     //Получаем и выводим на экран десять членов арифметической прогрессии
23     ai = a1;
24     cout << ai << endl;
25
26     ai = a1 + d;
27     cout << ai << endl;
28
29     ai = a1 + 2 * d;
30     cout << ai << endl;
31
32     ai = a1 + 3 * d;
33     cout << ai << endl;
34
35     ai = a1 + 4 * d;
36     cout << ai << endl;
37
38     ai = a1 + 5 * d;
39     cout << ai << endl;
40
41     ai = a1 + 6 * d;
42     cout << ai << endl;
43
44     ai = a1 + 7 * d;
45     cout << ai << endl;
46
47     ai = a1 + 8 * d;
48     cout << ai << endl;
49
50     ai = a1 + 9 * d;
51     cout << ai << endl;
52
53     //Долго и неудобно
54
55     return 0;
56 }
57
58 /*
59 * Случай 2. Приводим к виду, когда программный код повторяется,
60 * то есть к рекуррентному виду, когда значение переменных на следующем этапе
61 * зависят от значений переменных на предыдущем этапе
62 */
63
64 int main()
65 {
66     int a1 = 4;
67     int d = 3;
68
69     int ai;
70     int n = 10;
71
72     ai = a1;
```

```
73     cout << ai << endl;
74
75     ai = ai + d;
76     cout << ai << endl;
77
78     ai = ai + d;
79     cout << ai << endl;
80
81     ai = ai + d;
82     cout << ai << endl;
83
84     ai = ai + d;
85     cout << ai << endl;
86
87     ai = ai + d;
88     cout << ai << endl;
89
90     ai = ai + d;
91     cout << ai << endl;
92
93     ai = ai + d;
94     cout << ai << endl;
95
96     ai = ai + d;
97     cout << ai << endl;
98
99     ai = ai + d;
100    cout << ai << endl;
101
102    //Код повторяется, это удобнее, по крайней мере тем что его можно
103    //копировать - вставить
104
105    return 0;
106 }
107
108 /*
109  * Случай 3. заменяем повторяющийся код циклом
110  */
111 int main()
112 {
113     int a1 = 4;
114     int d = 3;
115
116     int ai;
117     int n = 10;
118
119     ai = a1;
120     /*
121     * Вводим дополнительную переменную - номер текущей итерации,
122     * то есть какой по номеру раз выполняется тело цикла
123     */
124     int i = 0;
125
126     while (i < n) //сравниваем не превысило ли число итераций максимальное
127     {
128         cout << ai << endl;
129         ai = ai + d;
130         i = i + 1; // увеличиваем номер итерации на один
131     }
132     //как только условие престанет выполняться цикл прекратится и выполнение
133     //перейдет к следующему за циклом оператору
134
135
136     return 0;
137 }
138
139 /*
140  * Случай 4. цикл for
141  */
142 int main()
143 {
144     int a1 = 4;
```

```
145     int d = 3;
146
147     int ai;
148     int n = 10;
149
150     ai = a1;
151
152     //замена цикла while на for
153     //int i = 0 и i = i+1 попадают в заголовок цикла
154     //и становятся слева и справа от условия соответственно
155
156     for (int i = 0; i < n; i = i + 1)
157     {
158         cout << ai << endl;
159         ai = ai + d;
160     }
161
162
163     return 0;
164 }
165
166
167
168 /*
169 * Случай 5. вывод всех элементов меньших 50
170 * В данном случае "счетчик итераций" не используется
171 * поэтому приводить цикл к виду for нецелесообразно
172 */
173 int main()
174 {
175     int a1 = 4;
176     int d = 3;
177
178     int ai;
179     int max = 50;
180
181     ai = a1;
182     while (ai < max)
183     {
184         cout << ai << endl;
185         ai = ai + d;
186     }
187
188
189     return 0;
190 }
191
192
193 /*
194 * Случай 6. Бесконечный цикл, выход при помощи "break"
195 */
196 int main()
197 {
198     int a1 = 4;
199     int d = 3;
200
201     int ai;
202     int max = 50;
203
204     ai = a1;
205     while (1)
206     {
207         cout << ai << endl;
208         if(ai >= max)
209             break;
210         ai = ai + d;
211     }
212
213
214     return 0;
215 }
```