

Побочные эффекты операции ввода и способы их устранения

Шаг 1. C-подмножество

Цель – с помощью цикла с постусловием осуществить ввод целого числа из диапазона значений [-128, 127].

Файл исходного кода – abnormal_input_C_1.cpp

```
/* Programming in C Style */

#include <stdio>

int main()
{
    int n; /* memory class - auto */

    printf("Programming in C style\n");
    do
    {
        printf("Number [-128, 127]? ");
        scanf("%d", &n);
    }
    while (!(n >= -128 && n <= 127));
    printf("%d\n", n);

    return 0;
}
```

Примечание:

Начиная с VisualStudio 2013, вместо **scanf()** следует использовать **scanf_s()**.

Протоколы испытаний

Укажем на одну из характерных ошибок проектирования программного кода:

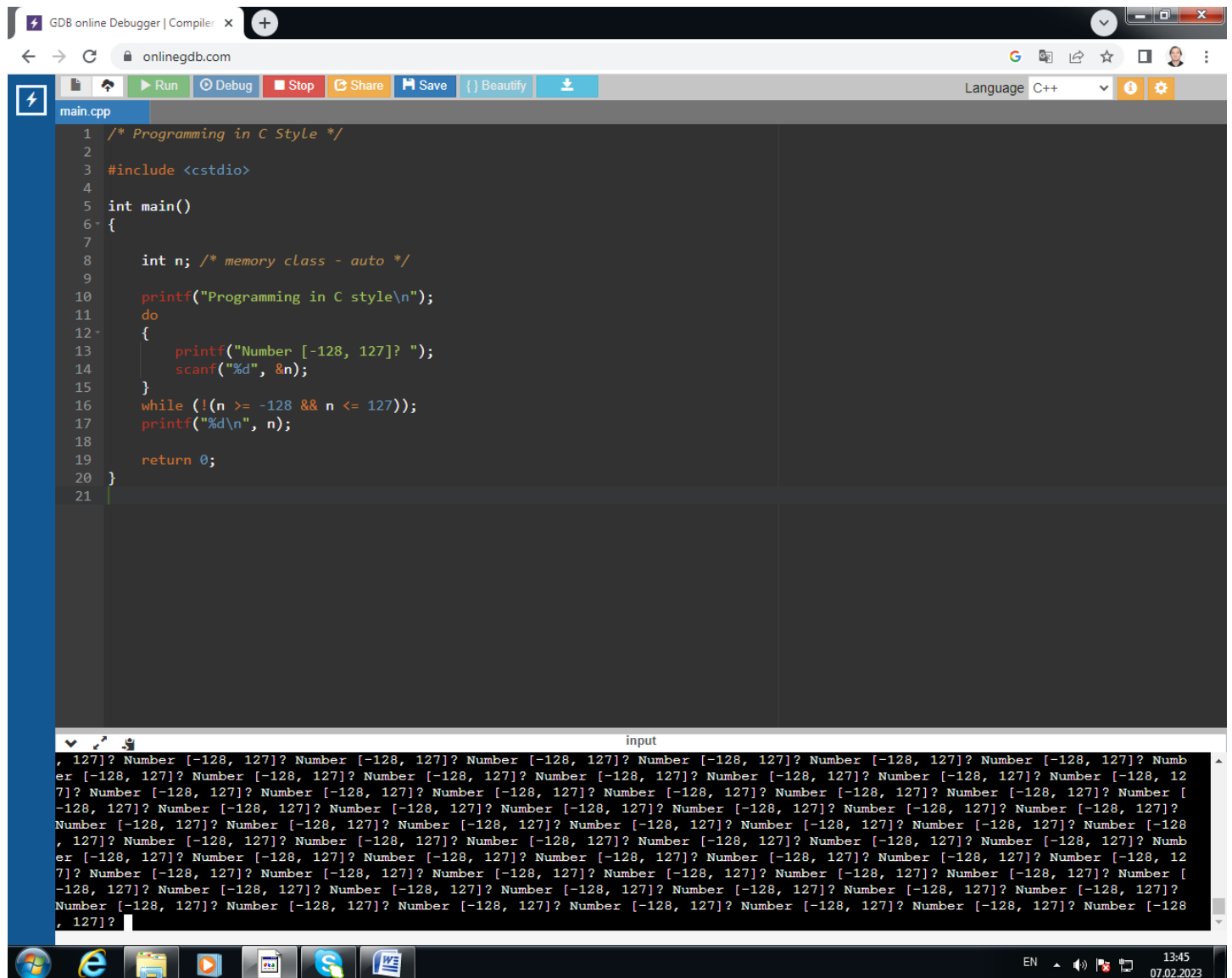
```
int n; /* memory class - auto */
```

Благодаря отсутствию инициализирующего значения при объявлении локальной переменной **n** (в данном контексте – это объект класса памяти **auto**) возможно проявление одного из двух побочных эффектов – в результате ввода значения “не число” конечный цикл с постусловием может стать бесконечным или в результате вывода значение переменной **n** может оказаться обескураживающим. Напомним, что локальные объекты с принадлежностью к классу памяти **auto** располагаются в стеке.

Станет ли цикл бесконечным или нет, станет ли результат операции вывода значения переменной **n** ожидаемым или нет – зависит от состояния ячеек автоматической памяти (стека), в которых располагалась эта переменная.

В качестве полигона для испытаний обратимся к VisualStudio2019 и к одному из семейства онлайн-компиляторов – OnlineGDB.

Неблагополучный исход при вводе значения “не число”.



The screenshot displays the OnlineGDB web interface. The top navigation bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The code editor shows a C++ program named `main.cpp` with the following content:

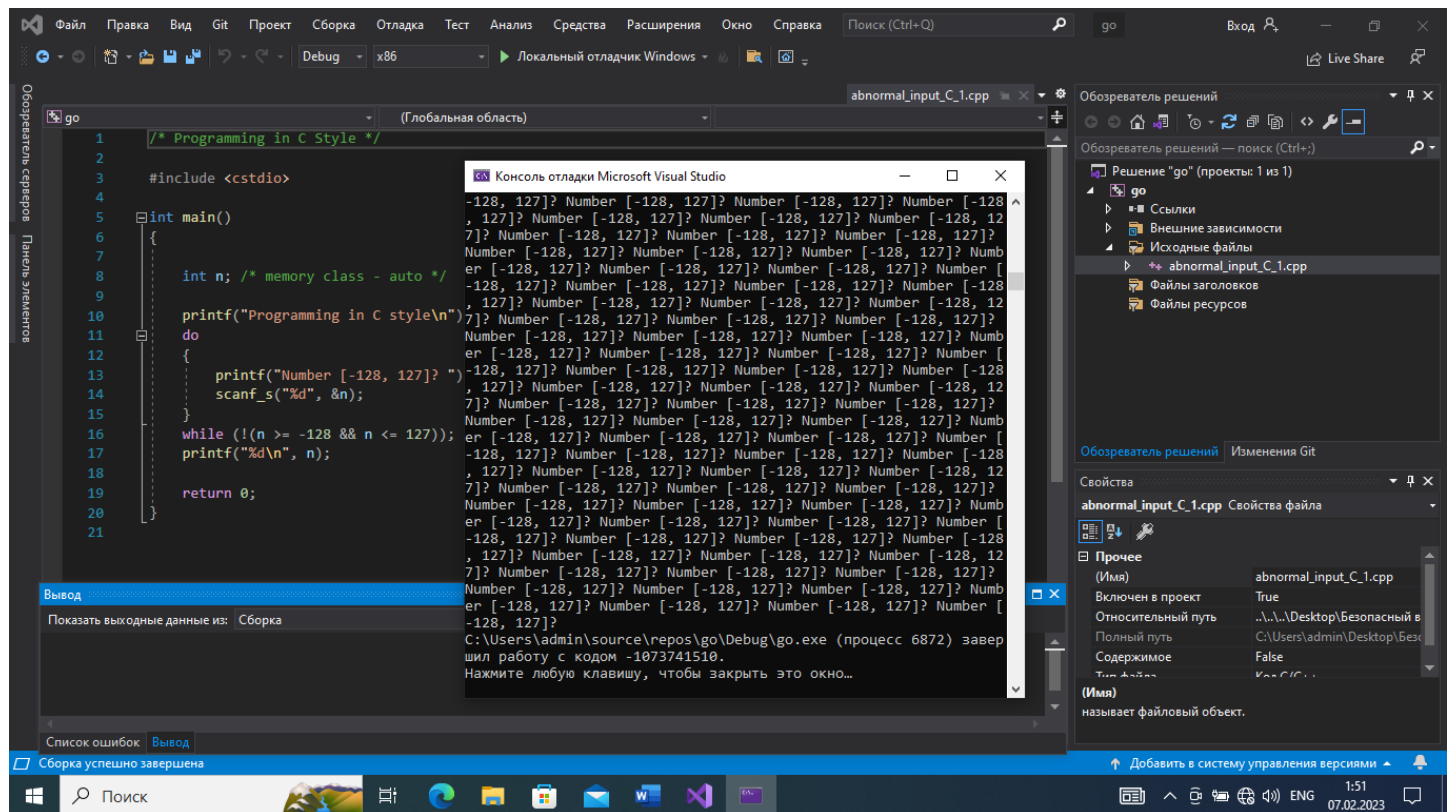
```
1 /* Programming in C Style */
2
3 #include <stdio>
4
5 int main()
6 {
7
8     int n; /* memory class - auto */
9
10    printf("Programming in C style\n");
11    do
12    {
13        printf("Number [-128, 127]? ");
14        scanf("%d", &n);
15    }
16    while (!(n >= -128 && n <= 127));
17    printf("%d\n", n);
18
19    return 0;
20 }
21
```

The output window at the bottom shows the execution results. It displays a repeating pattern of the prompt "Number [-128, 127]?" followed by a question mark, indicating that the program is stuck in an infinite loop because the input is not a valid integer within the specified range.

Как видим, в результате ввода значения “не число” конечный цикл с постусловием стал бесконечным. Причина проявления такого рода побочного эффекта очевидна – операция ввода привела к ошибке форматного преобразования символов вводимой строки в целое число, что в итоге позволило оставить неизменным первоначальное значение в виде “целочисленного мусора” при объявлении переменной `n`, которое изначально не принадлежало диапазону значений `[-128, 127]`. Тем самым, это обстоятельство как раз и послужило триггером для запуска бесконечного цикла.

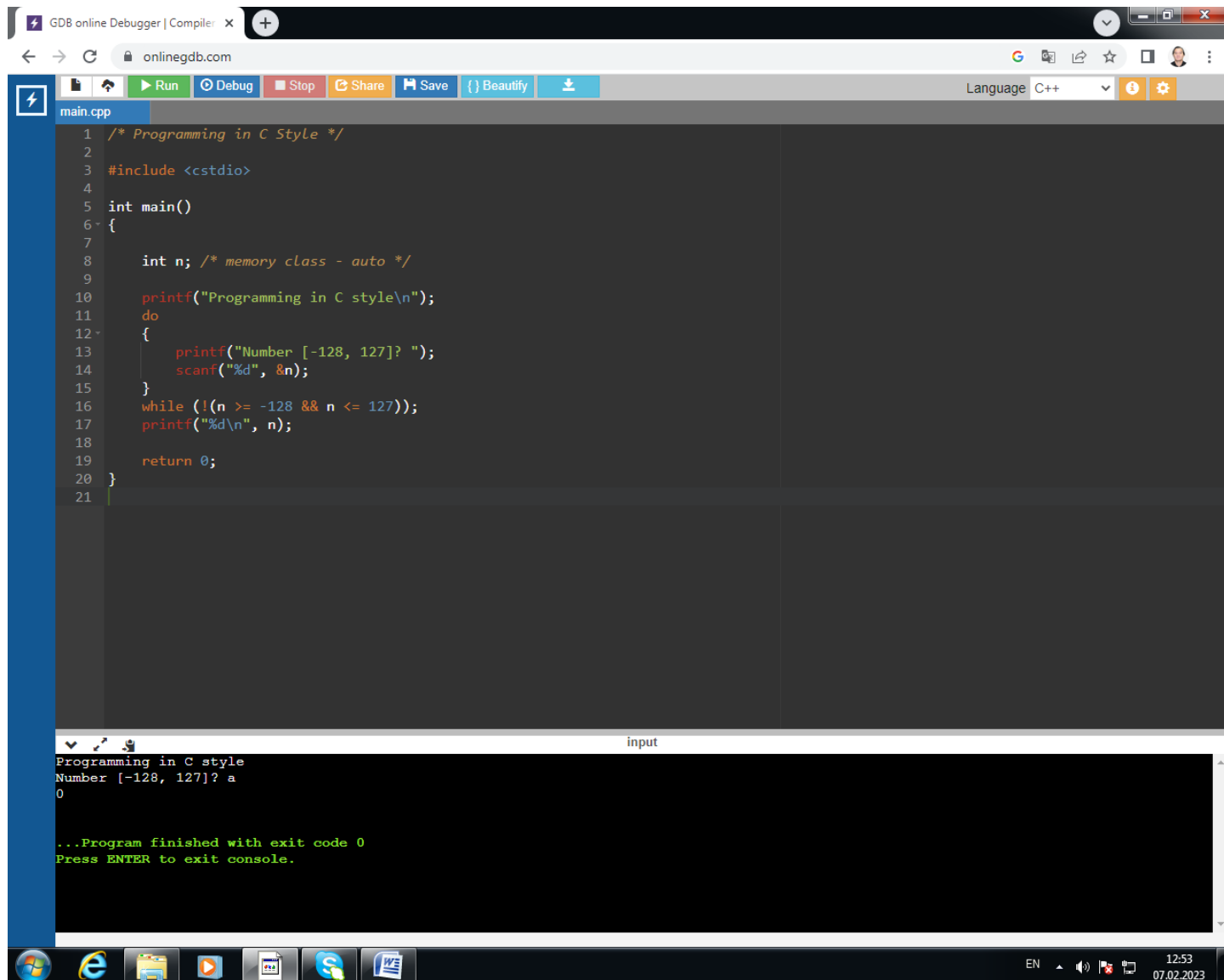
Visual Studio 2019

Неблагополучный исход при вводе значения “не число”.



OnlineGDB

Неблагополучный исход при вводе значения “не число”.



The screenshot shows the OnlineGDB web interface. The top bar includes the site name 'GDB online Debugger | Compiler' and the URL 'onlinegdb.com'. Below the bar is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The main area is divided into two panes. The left pane shows the source code for 'main.cpp':

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7
8      int n; /* memory class - auto */
9
10     printf("Programming in C style\n");
11     do
12     {
13         printf("Number [-128, 127]? ");
14         scanf("%d", &n);
15     }
16     while (!(n >= -128 && n <= 127));
17     printf("%d\n", n);
18
19     return 0;
20 }
21
```

The right pane shows the execution output in a terminal window titled 'input':

```
Programming in C style
Number [-128, 127]? a
0

...Program finished with exit code 0
Press ENTER to exit console.
```

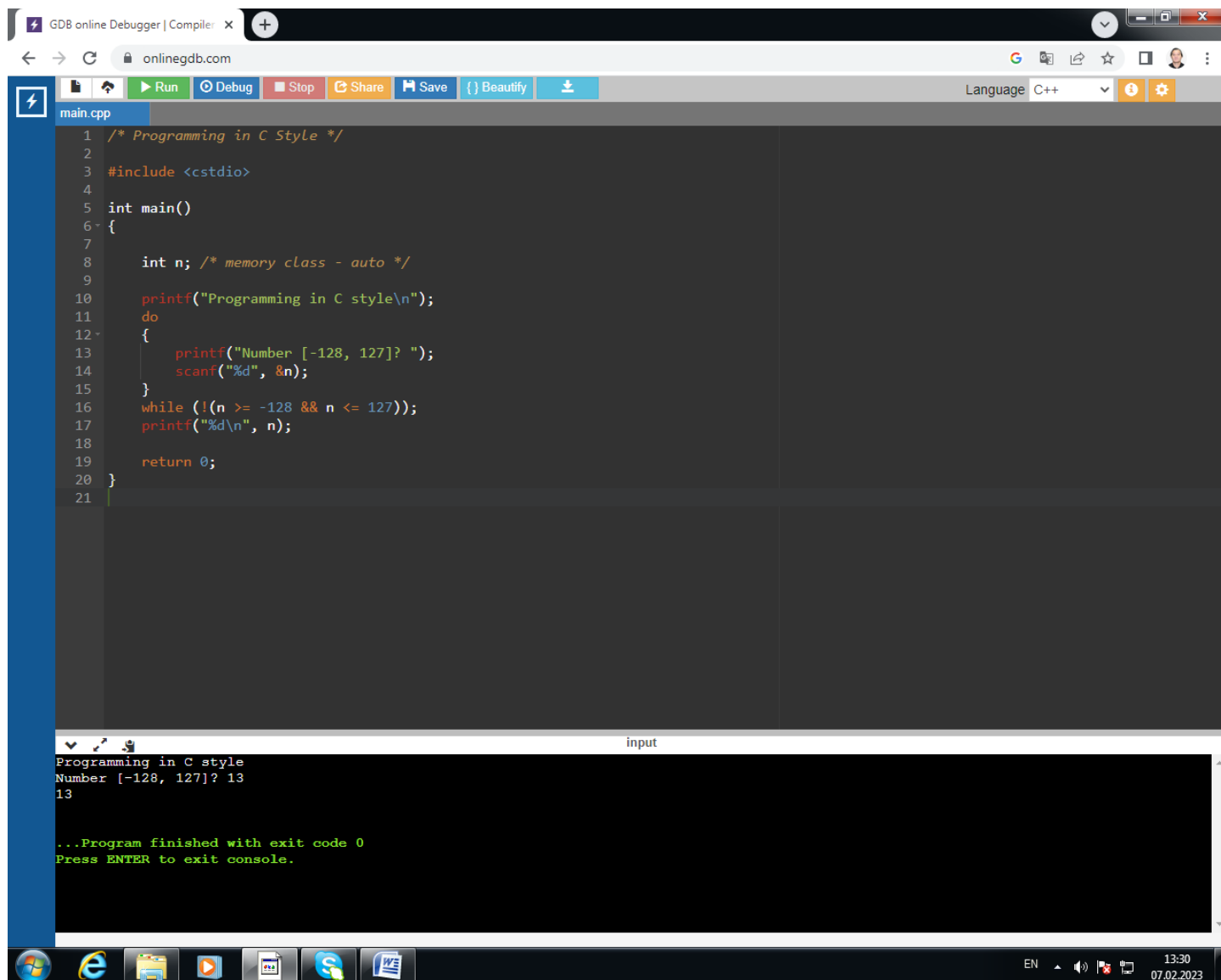
The bottom of the image shows a Windows taskbar with various application icons and a system clock displaying 12:53 on 07.02.2023.

Как видим, в результате ввода значения “не число” конечный цикл с постусловием на этот раз не стал бесконечным. При этом результат операции вывода значения переменной *n* подтвердил предположение об одном из приемлемых первоначальных значений в виде “целочисленного мусора” при её объявлении, которое изначально принадлежало диапазону значений [-128, 127].

Отметим также, что операция ввода, как и следовало ожидать, привела к ошибке форматного преобразования символов вводимой строки в целое число, что позволило оставить неизменным первоначальное значение для переменной *n*.

OnlineGDB

Ожидаемый благополучный исход при вводе значения “число”.



The screenshot displays the OnlineGDB web interface. The top navigation bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C++. The main editor shows a C++ program in `main.cpp` with the following code:

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7
8      int n; /* memory class - auto */
9
10     printf("Programming in C style\n");
11     do
12     {
13         printf("Number [-128, 127]? ");
14         scanf("%d", &n);
15     }
16     while (!(n >= -128 && n <= 127));
17     printf("%d\n", n);
18
19     return 0;
20 }
21
```

The console output shows the program's execution:

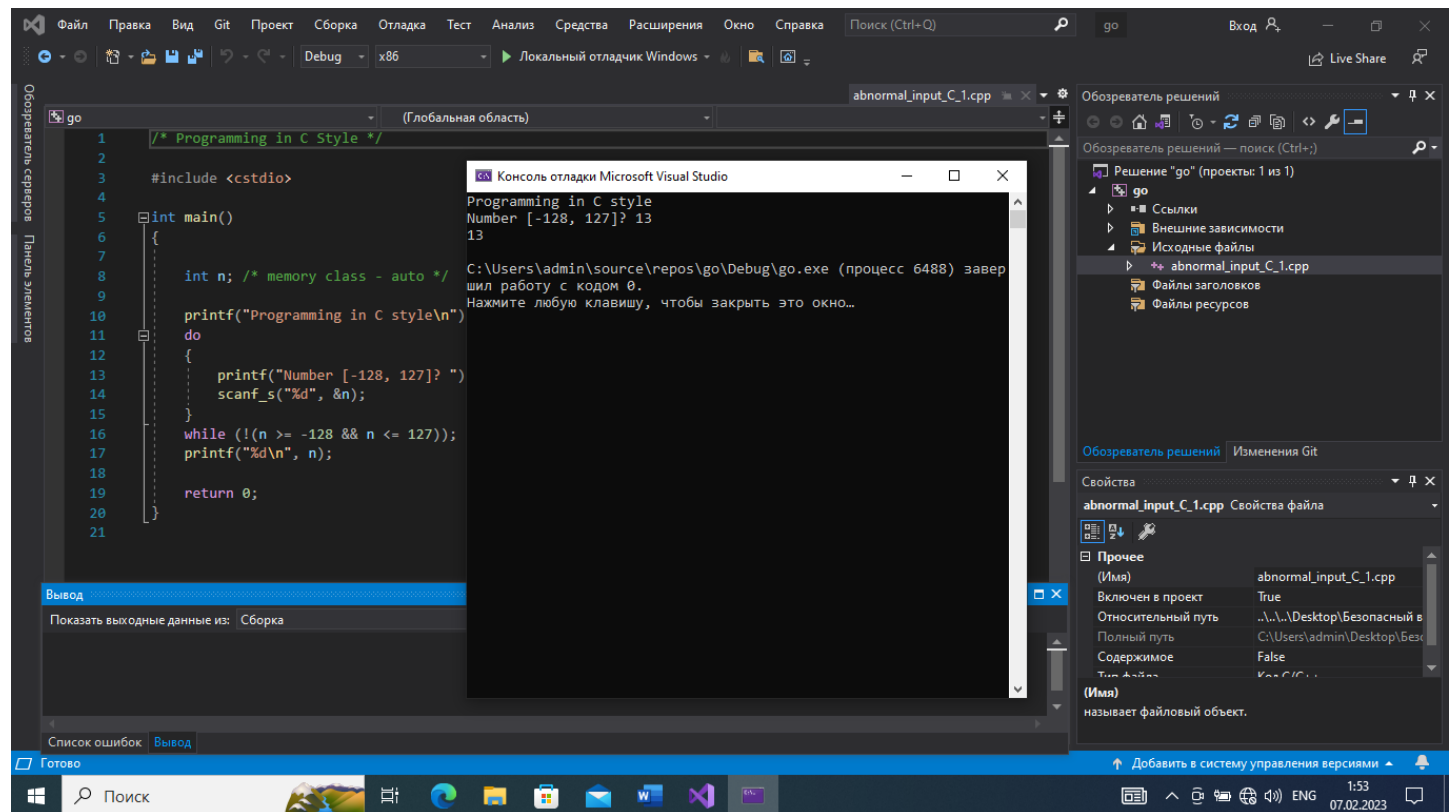
```
Programming in C style
Number [-128, 127]? 13
13

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom shows the system clock as 13:30 on 07.02.2023.

Visual Studio 2019

Ожидаемый благополучный исход при вводе значения “число”.



Шаг 2. C-подмножество

Цель – с помощью цикла с постусловием осуществить ввод целого числа из диапазона значений $[-128, 127]$ с подтверждением или отказом в осуществлении намерения в повторной операции ввода.

Файл исходного кода – abnormal_input_C_2.cpp

```
/* Programming in C Style */

#include <stdio>

int main()
{

    int n; /* memory class - auto */
    char answer; /* memory class - auto */

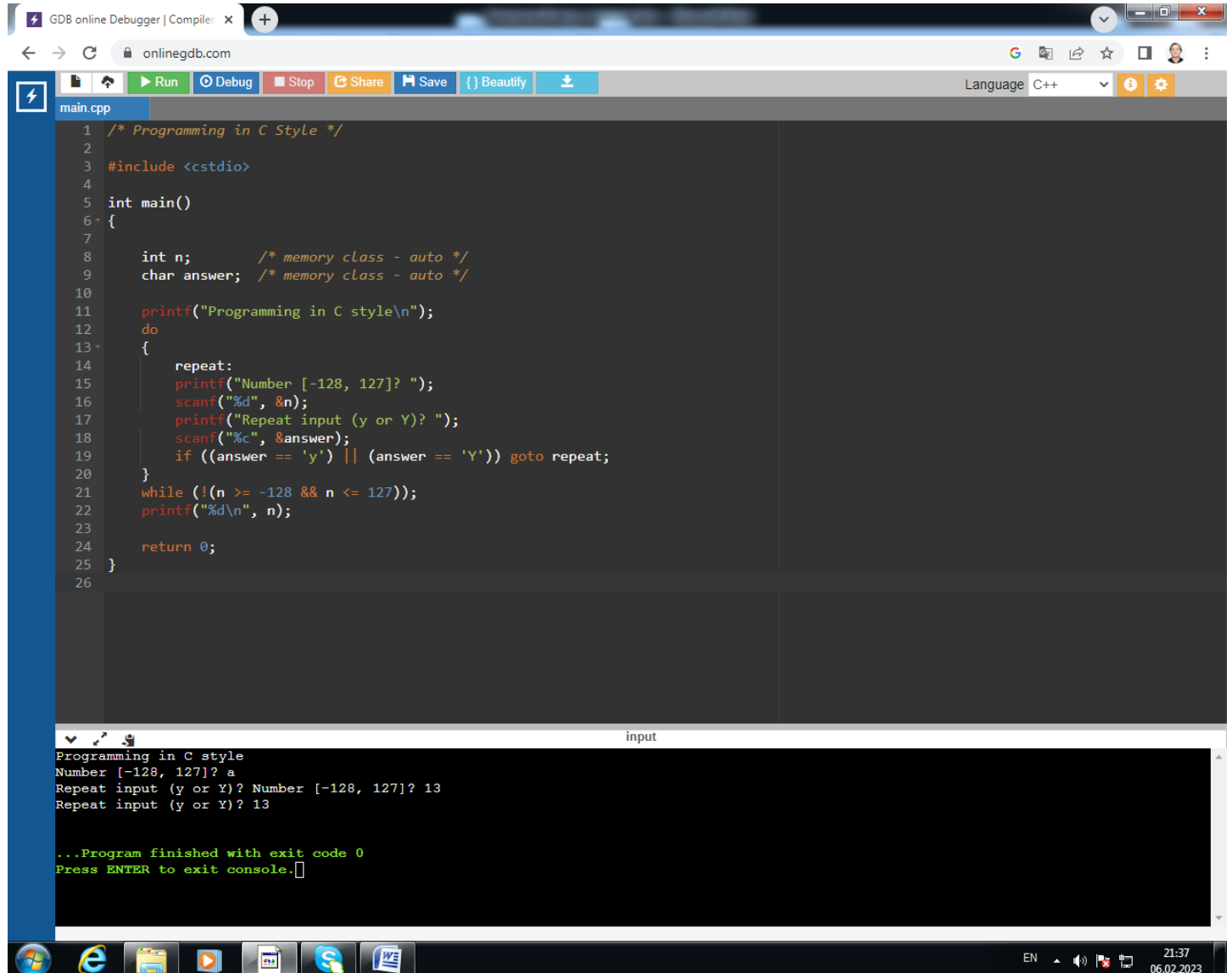
    printf("Programming in C style\n");
    do
    {
        repeat:
        printf("Number [-128, 127]? ");
        scanf("%d", &n);
        printf("Repeat input (y or Y)? ");
        scanf("%c", &answer);
        if ((answer == 'y') || (answer == 'Y')) goto repeat;
    }
    while (!(n >= -128 && n <= 127));
    printf("%d\n", n);

    return 0;
}
```


Протоколы испытаний

OnlineGDB

Неблагополучный исход при вводе значения “не число” с непреднамеренным отказом в осуществлении намерения в повторной операции ввода.



The screenshot displays the OnlineGDB web interface. The top section shows the source code for a C program named `main.cpp`. The code is as follows:

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7
8      int n;          /* memory class - auto */
9      char answer;    /* memory class - auto */
10
11     printf("Programming in C style\n");
12     do
13     {
14         repeat:
15         printf("Number [-128, 127]? ");
16         scanf("%d", &n);
17         printf("Repeat input (y or Y)? ");
18         scanf("%c", &answer);
19         if ((answer == 'y') || (answer == 'Y')) goto repeat;
20     }
21     while (!(n >= -128 && n <= 127));
22     printf("%d\n", n);
23
24     return 0;
25 }
26
```

The bottom section shows the execution output in a terminal window. The output is as follows:

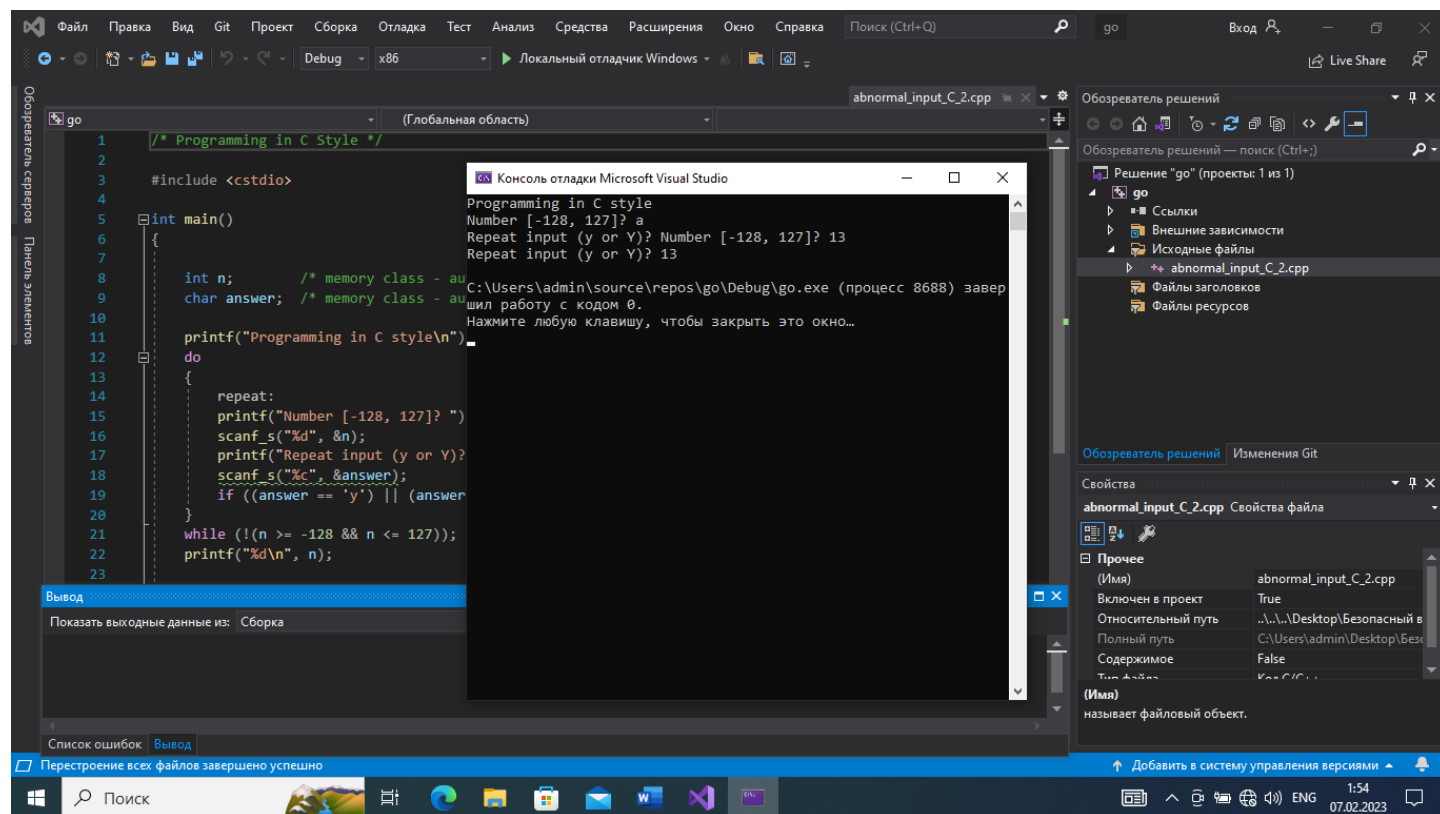
```
Programming in C style
Number [-128, 127]? a
Repeat input (y or Y)? Number [-128, 127]? 13
Repeat input (y or Y)? 13

...Program finished with exit code 0
Press ENTER to exit console.
```

The interface also includes a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The language is set to C++.

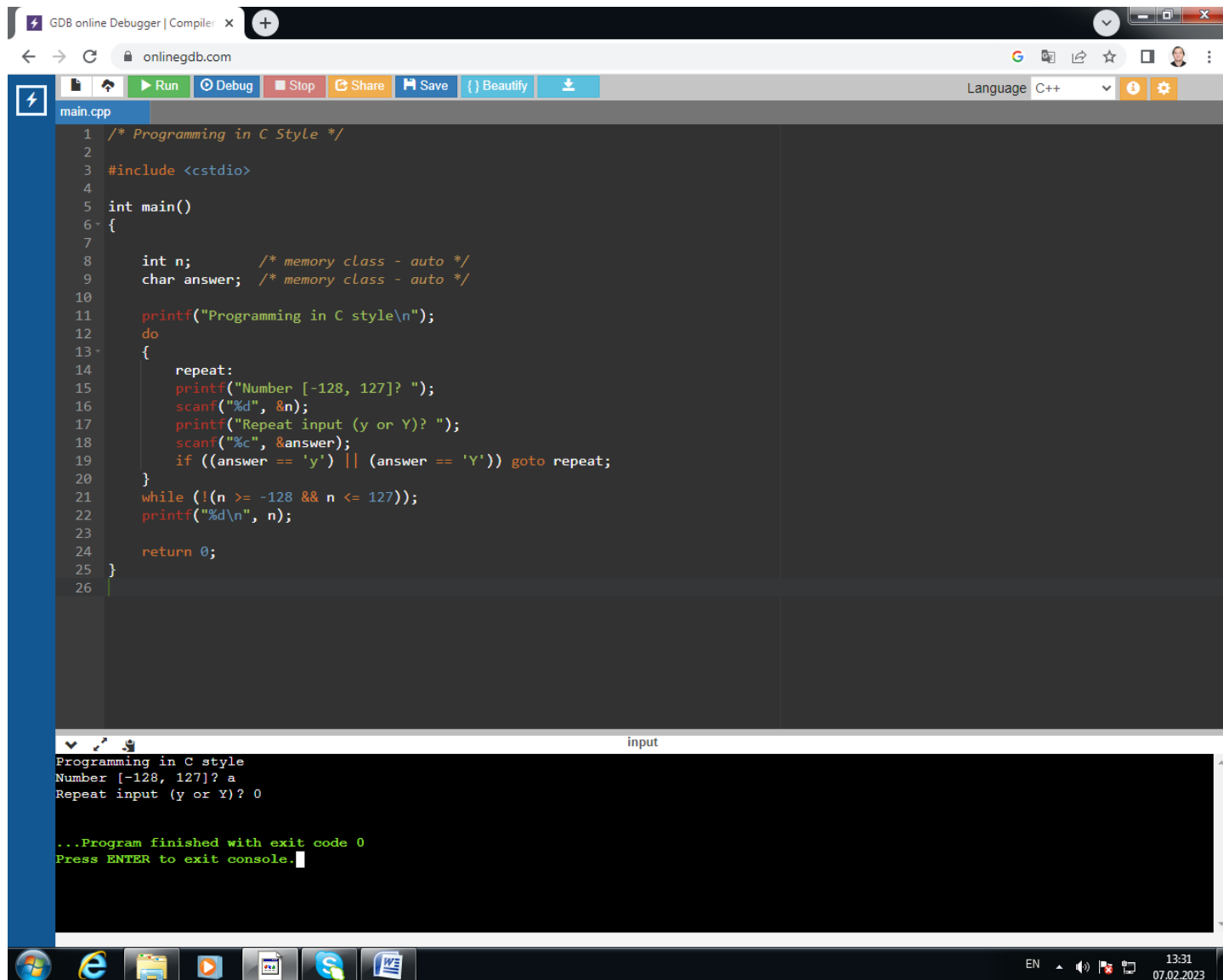
Visual Studio 2019

Неблагополучный исход при вводе значения “не число” с непреднамеренным отказом в осуществлении намерения в повторной операции ввода.



OnlineGDB

Неблагополучный исход при вводе значения “не число” с непреднамеренным отказом в осуществлении намерения в повторной операции ввода.



The screenshot displays the OnlineGDB web interface. The top navigation bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The main editor shows a C program named main.cpp. The program prompts the user for a number and a confirmation to repeat. The execution output at the bottom shows the program's behavior when the input 'a' is provided, which is not a valid integer, leading to an exit code of 0.

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7
8      int n;          /* memory class - auto */
9      char answer;    /* memory class - auto */
10
11     printf("Programming in C style\n");
12     do
13     {
14         repeat:
15         printf("Number [-128, 127]? ");
16         scanf("%d", &n);
17         printf("Repeat input (y or Y)? ");
18         scanf("%c", &answer);
19         if ((answer == 'y') || (answer == 'Y')) goto repeat;
20     }
21     while (!(n >= -128 && n <= 127));
22     printf("%d\n", n);
23
24     return 0;
25 }
26
```

input

```
Programming in C style
Number [-128, 127]? a
Repeat input (y or Y)? 0

...Program finished with exit code 0
Press ENTER to exit console.
```

Шаг 3. С-подмножество

Цель – после завершения операции ввода целого числа преднамеренно осуществить ввод символа с помощью цикла с постусловием до тех пор, пока из буфера ввода не будет считан символ LF – “новая строка”.

Файл исходного кода – input_C_1.cpp

```
/* Programming in C Style */

#include <stdio>

int main()
{
    char c;
    int n = 0; /* default value */

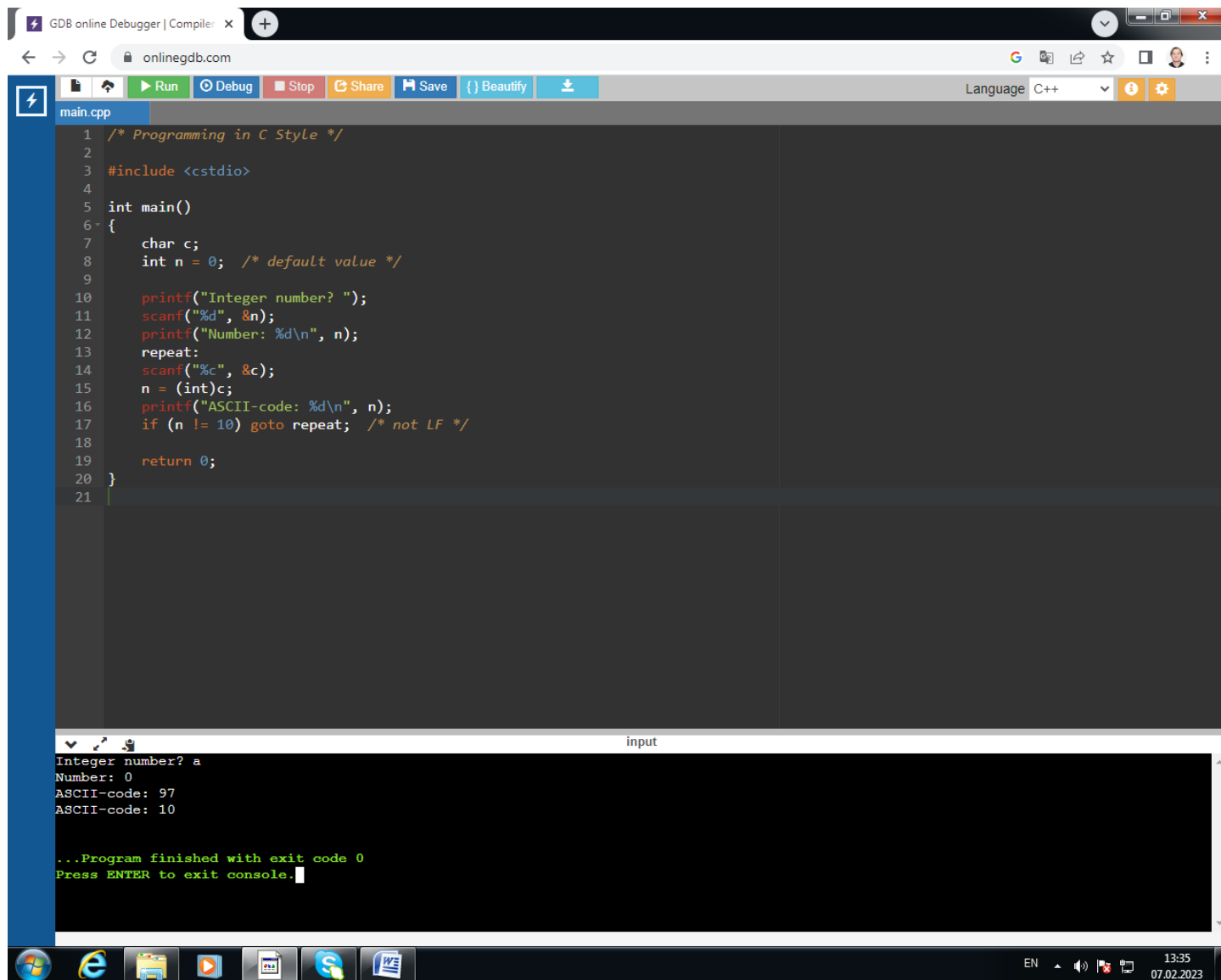
    printf("Integer number? ");
    scanf("%d", &n);
    printf("Number: %d\n", n);
    repeat:
    scanf("%c", &c);
    n = (int)c;
    printf("ASCII-code: %d\n", n);
    if (n != 10) goto repeat; /* not LF */

    return 0;
}
```

Протоколы испытаний

OnlineGDB

Неблагополучный исход при вводе значения “не число”.



The screenshot displays the OnlineGDB web interface. The top section shows the source code for a C++ program named `main.cpp`. The code prompts the user for an integer, reads it, and then prints its ASCII code. It includes a loop to handle non-integer input by repeatedly asking for input until a valid integer is entered. The bottom section shows the program's execution output, which matches the expected behavior for the input 'a'.

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7      char c;
8      int n = 0; /* default value */
9
10     printf("Integer number? ");
11     scanf("%d", &n);
12     printf("Number: %d\n", n);
13     repeat:
14     scanf("%c", &c);
15     n = (int)c;
16     printf("ASCII-code: %d\n", n);
17     if (n != 10) goto repeat; /* not LF */
18
19     return 0;
20 }
21
```

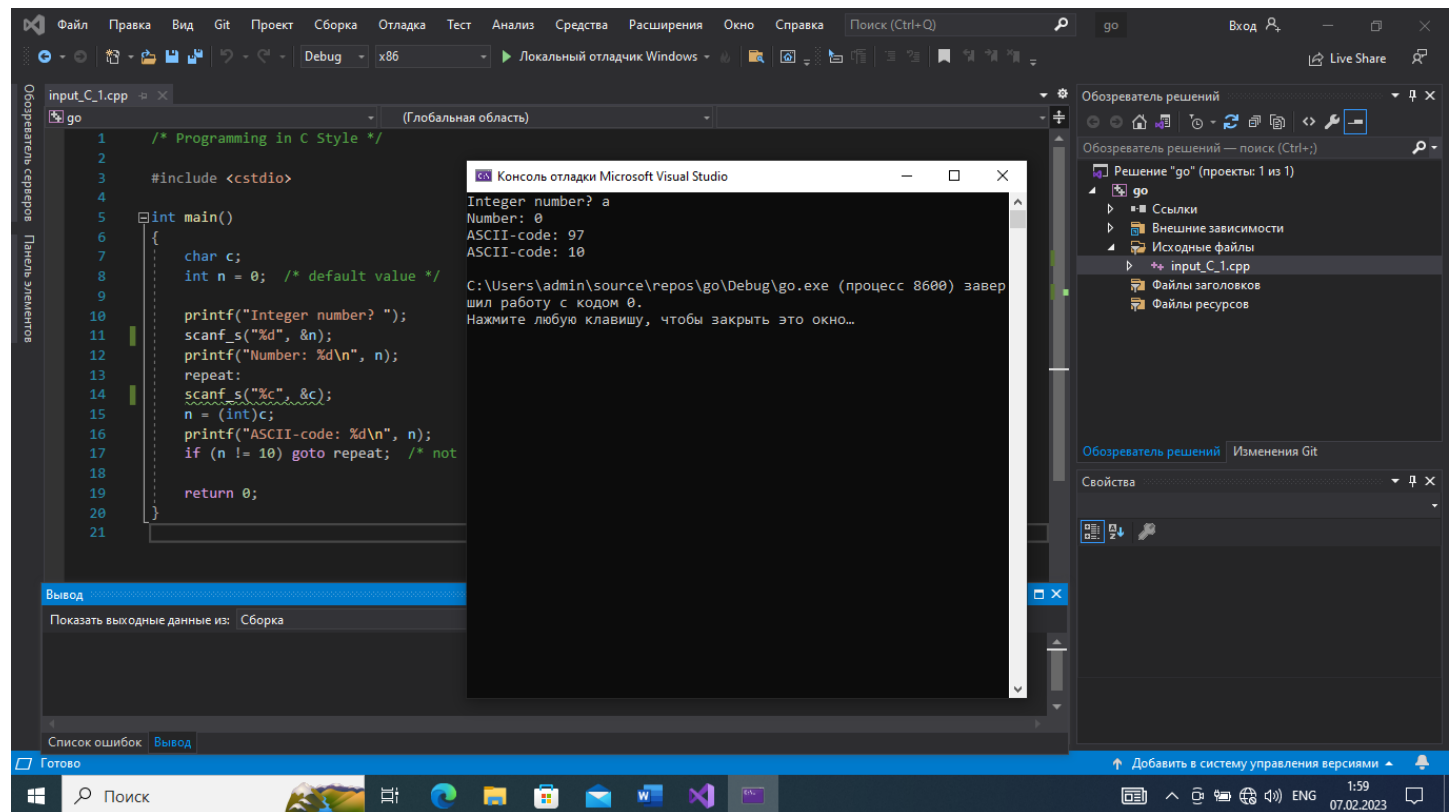
input

```
Integer number? a
Number: 0
ASCII-code: 97
ASCII-code: 10

...Program finished with exit code 0
Press ENTER to exit console.
```

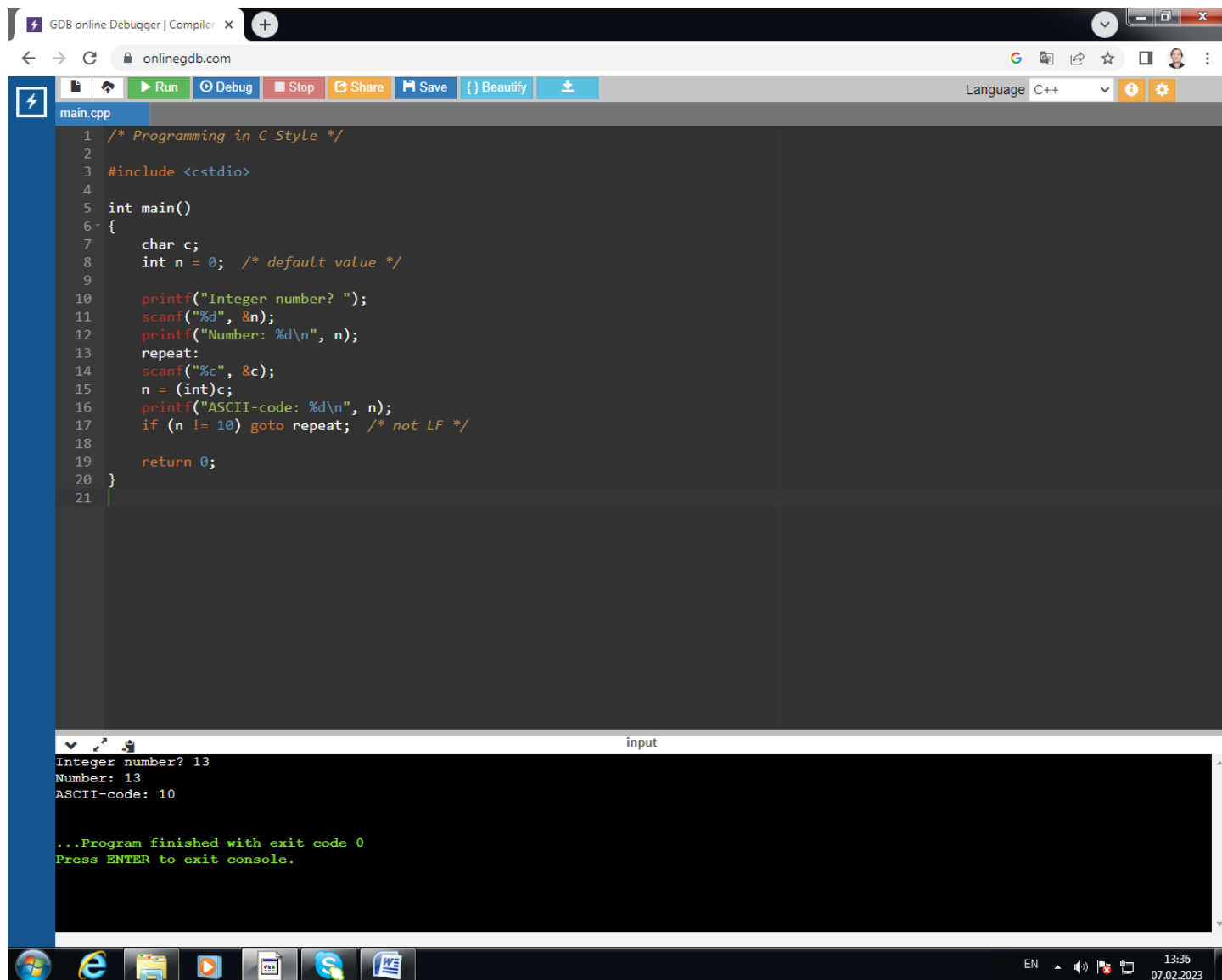
Visual Studio 2019

Неблагополучный исход при вводе значения “не число”.



OnlineGDB

Благополучный исход при вводе значения “число”.



The screenshot displays the OnlineGDB web interface. The top navigation bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The main editor shows a C++ program in `main.cpp` that prompts for an integer, reads it, and prints its ASCII code. The console output shows the program finished with exit code 0.

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7      char c;
8      int n = 0; /* default value */
9
10     printf("Integer number? ");
11     scanf("%d", &n);
12     printf("Number: %d\n", n);
13     repeat:
14     scanf("%c", &c);
15     n = (int)c;
16     printf("ASCII-code: %d\n", n);
17     if (n != 10) goto repeat; /* not LF */
18
19     return 0;
20 }
21
```

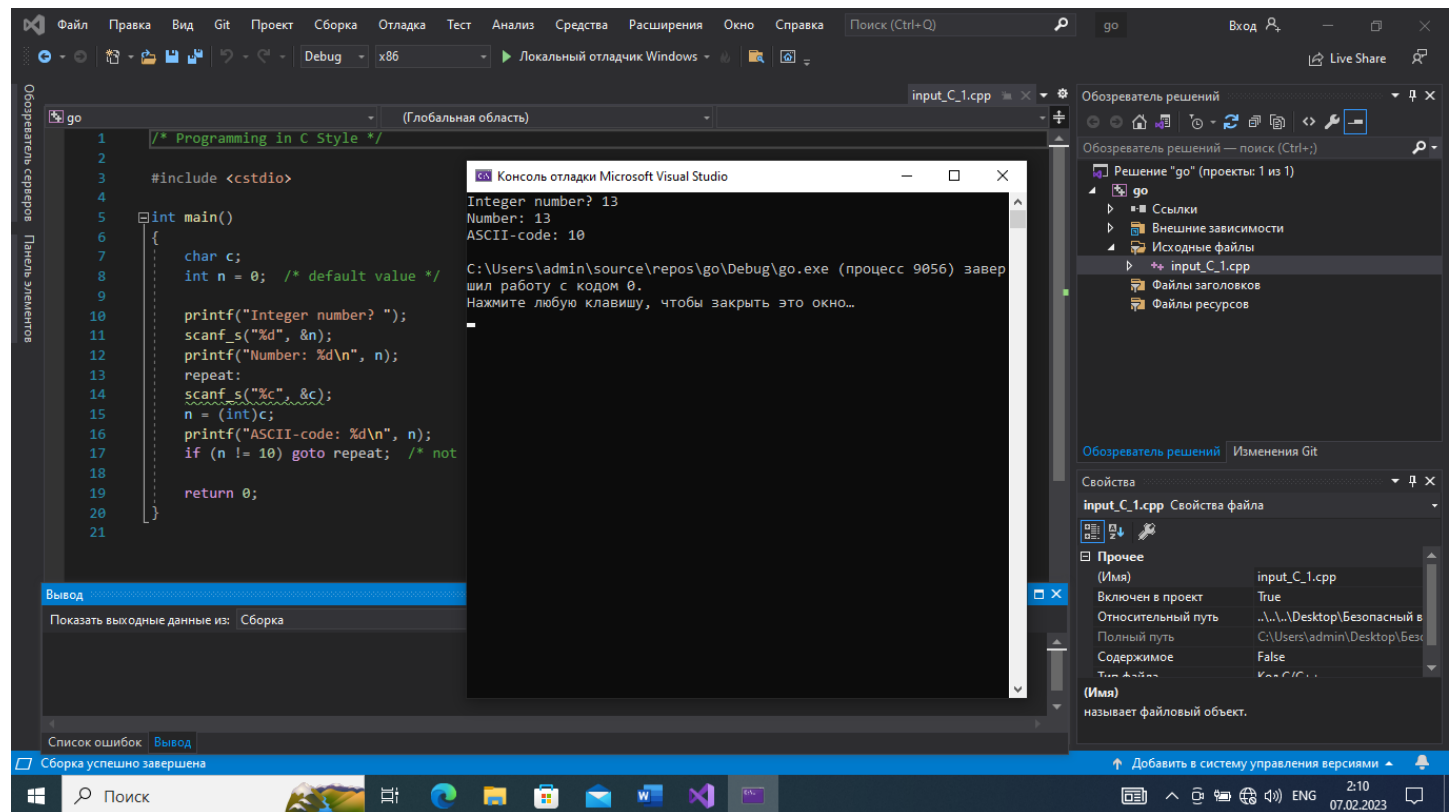
input

```
Integer number? 13
Number: 13
ASCII-code: 10

...Program finished with exit code 0
Press ENTER to exit console.
```

Visual Studio 2019

Благополучный исход при вводе значения “число”.



Шаг 4. C-подмножество

Цель – продолжая исследование содержимого буфера ввода после завершения операции ввода целого числа, осуществить ввод символа с помощью функции **getchar()**.

Файл исходного кода – input_C_2.cpp

```
/* Programming in C Style */

#include <stdio>

int main()
{
    char c;
    int n = 0; /* default value */

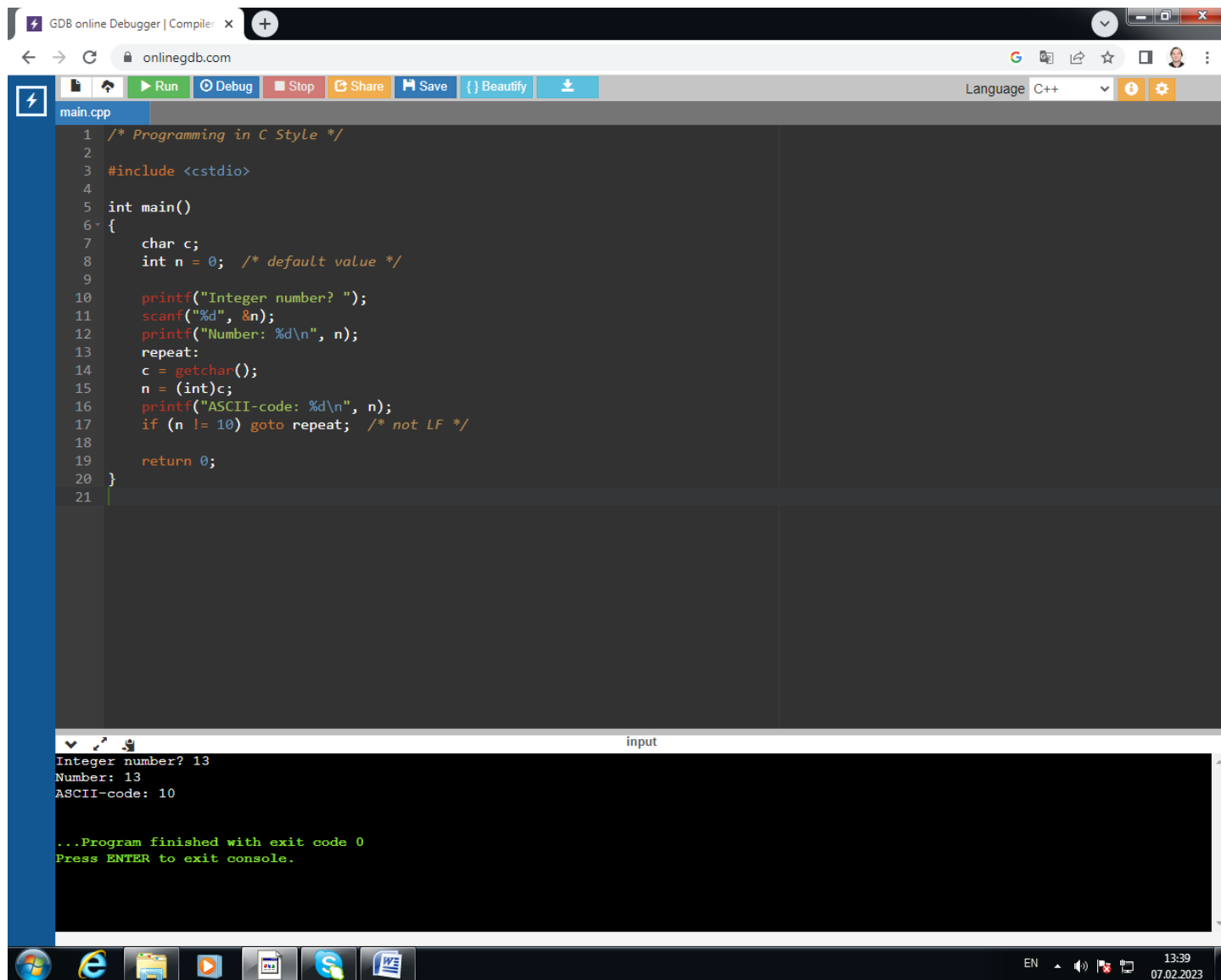
    printf("Integer number? ");
    scanf("%d", &n);
    printf("Number: %d\n", n);
    repeat:
    c = getchar();
    n = (int)c;
    printf("ASCII-code: %d\n", n);
    if (n != 10) goto repeat; /* not LF */

    return 0;
}
```

Протоколы испытаний

OnlineGDB

Благополучный исход при вводе значения “число”.



The screenshot displays the OnlineGDB web interface. The top bar shows the browser address bar with 'onlinegdb.com' and a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The main area shows a C++ program in 'main.cpp' with the following code:

```
1  /* Programming in C Style */
2
3  #include <stdio>
4
5  int main()
6  {
7      char c;
8      int n = 0; /* default value */
9
10     printf("Integer number? ");
11     scanf("%d", &n);
12     printf("Number: %d\n", n);
13     repeat:
14     c = getchar();
15     n = (int)c;
16     printf("ASCII-code: %d\n", n);
17     if (n != 10) goto repeat; /* not LF */
18
19     return 0;
20 }
21
```

The bottom panel shows the program's output in a console window titled 'input':

```
Integer number? 13
Number: 13
ASCII-code: 10

...Program finished with exit code 0
Press ENTER to exit console.
```

The Windows taskbar at the bottom shows the system clock as 13:39 on 07.02.2023.

Visual Studio 2019

Благополучный исход при вводе значения “число”.

