

3.

The screenshot shows the GDB Online interface. The top window displays the source code 'main.c' with line numbers 1 through 20. The code includes a welcome message, imports stdio.h, defines variables a=10, b=5, c=2, x=1, y=3, z=4, and calculates result = (a+x+y)-c. It then prints the values of a, b, c, x, y, and z. The bottom window shows the terminal output: 'result = 48', followed by the variable assignments 'a = 10, b = 5, c = 2, x = 2, y = 2, z = 4'. The message '...Program finished with exit code 0' is displayed, along with the instruction 'Press ENTER to exit console.'

```
main.c
1  ****
2
3  Welcome to GDB Online.
4  GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby,
5  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
6  Code, Compile, Run and Debug online from anywhere in world.
7
8  ****
9 # include <stdio.h>
10
11 int main () {
12     int a = 10, b = 5, c = 2, x = 1, y = 3, z = 4, result ;
13
14     // Construct your expression here
15     result = a * (++x + y--) - c;
16     printf (" result = %d\n", result );
17     printf ("a = %d, b = %d, c = %d, x = %d, y = %d, z = %d\n", a, b, c, x, y, z);
18
19     return 0;
20 }
```

input

```
result = 48
a = 10, b = 5, c = 2, x = 2, y = 2, z = 4

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

4.

- a. The output of result in the program is 11.692308.
- b. The calculation in the program is $(a + b) / --c$. Here, a is an integer, b and c are floating-point numbers, and $--c$ is a pre-decrement operation.
In the pre-decrement operation $--c$, the value of c is decremented by 1 before it is used in the expression. So, c which was initially 2.3 becomes 1.3 after the pre-decrement operation.
When an operation involves both integer and floating-point numbers, the integer is automatically converted to a floating-point number before the operation is performed.
So, the calculation becomes $(10.0 + 5.2) / 1.3$, which equals $15.2 / 1.3$, and the result is 11.692308, due to the precision limit of the float data type in C, the output is rounded and printed as 11.692308 when displayed with the `%f` format specifier in the `printf` function.