Assignment 1 – Evaluating Expressions (5 points)

(i)  $\frac{n}{f} \times d$  
(ii)  $u == 4 \times ++d$  
(iii)  $0 < d < 2 || 2 > f$  
(iv)  $0 < 1 && 1 != 1 || ++x > 3$  
(v)  $3 \times d > d || 1 / z != 0 && 3 + 4 >= 7$

For each one, state the type and value! You should solve this exercise without the help of Codeboard, just as you are asked to do in exams. Once you computed the results yourself, you can of course check by implementing the expressions in Codeboard. If the results don’t match, try to find the mistake. In the beginning of each sub-task, assume the following initializations:

```c
int d = 2; int u = 4;
int f = 5; int x = 3;
int n = 9; int z = 0;
```

Remark: When you are writing your own programs, you should usually avoid writing too complex expressions. They can make your code unreadable and it’s easy to get well-hidden mistakes. Some of the above expressions should indicate this!

Assignment 2 – Cross Sum (4 points)

Write a program `cross_sum.cpp` that inputs a natural number $n$ (including 0) and outputs the sum of the (decimal) digits of $n$.

I/O-Examples

```
652
Cross sum of 652 is: 13

86400
Cross sum of 86400 is: 18
```

Submission:  [https://codeboard.ethz.ch/ifmp16E3T2](https://codeboard.ethz.ch/ifmp16E3T2)
Assignment 3 – Reverse Binary Representation (4 points)

Write a program `dec2bin.cpp` that inputs a natural number \( n \) (including 0) and outputs the binary digits of \( n \) in reverse order.

### I/O-Examples

<table>
<thead>
<tr>
<th>Input</th>
<th>Reverse Binary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>12</td>
<td>0011</td>
</tr>
</tbody>
</table>

Submission: [https://codeboard.ethz.ch/ifmp16E3T3](https://codeboard.ethz.ch/ifmp16E3T3)

Challenge - Even Connected Subsequences (8 points)

This week’s challenge is exercise 53 from the script. In addition to what the exercise description says, you can assume that the user enters 2 to indicate the end of the sequence.

To implement your program, please refer to the description on how you can write your own programs.