Lab 8
Deadline: In lab on Nov 23

Implement a C program that performs addition/subtraction without using the + or - operators, detects overflow within the specified bitwidth, and performs right shifting. The program should take command line arguments for bitwidth, two numbers to be added, and a right shift number. In case of overflow, the program should print a message overflow detected within the specified bitwidth and perform right shifting with the provided number.

Requirements

- Implement a function **logicaladder** that adds two numbers without using the + or – operators and returns result.
- Implement a function **detectoverflow** to check if overflow occurs within the specified bitwidth. If no overflow occurs, the function should return the result of addition. If overflow occurs, the function should perform right shift on result of addition by provided shift number and return right shift result.
- Print to the console that overflow has occurred within the specified bitwidth.

Implementation Details

- The program should take exactly 4 command line arguments:
  Bitwidth (8, 16, 32, or 64), Number 1, Number 2, Shift Number.
- Validate that the provided bitwidth is one of the supported values: 8, 16, 32, or 64.
- The addition without using + or - should be implemented using bitwise operations.
- Overflow detection should be based on the specified bitwidth.
- The right shifting operation should be performed on the result after addition, if overflow occurs.

Restrictions

- You cannot use special cases based on bitwidths, whether the values of bitwidth is 8, 16, 32 or 64, your program must use a unified code path to handle all values of bitwidths without special case handling.

Output

- Display the result of the addition.
- If overflow occurs, Display the result after shifting.
- If overflow occurs, indicate that overflow has occurred within the specified bitwidth.

How to Compile and Run

Compile the program using a C compiler (e.g., gcc):

```
gcc lab8.c -o < executable>
./<executable> 8 102 5 2
```

Expected Output:

```
Result of addition: 107
```

Run:

```
./<executable> 8 125 5 1
```

Expected Output:

```
Result after right shift: 65
Overflow detected within the specified bitwidth.
```

Grading

- (1 point) Correct implementation of addition.
- (1 point) Proper overflow detection and reporting.
- (1 point) Dynamic handling of different bitwidths.
- (1 point) Correct implementation of right shifting

Submission Files

- You must submit only one file named to Learning Hub: **lab8.c**
- Submit it to learning hub before the deadline