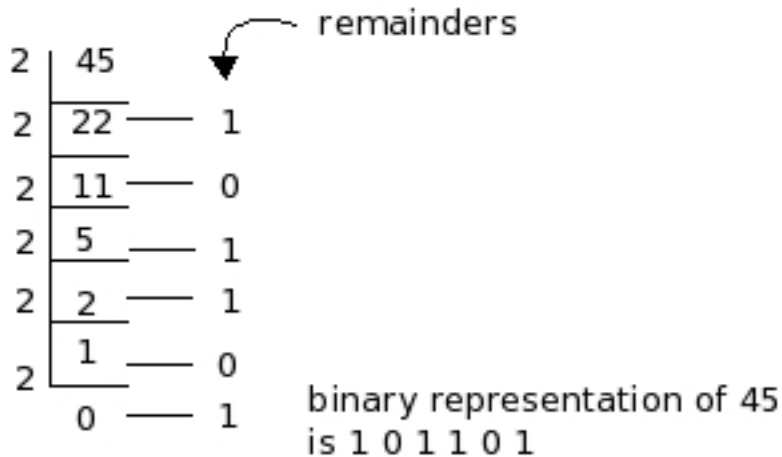


## Exercises on Stacks

1. Converting a decimal integer into binary can be implemented by a stack data structure. The arithmetic involved is as follows:



You could use the % (mod) arithmetic operator to find the remainders, push the remainders onto a stack and then pop the stack repeatedly to get the binary representation of the given decimal integer. Write a C++ program that accepts an integer as input and prints out its binary representation.

2. Write a program to check that the opening and closing braces in a line of user input matches. Use a stack. `ab{cdef23{wewew}sdfjfkfjds}dfsfsdf` produces a match, while `ab{cdef23{wewew}sdfj}kfkfjds}dfsfsdf` doesn't.

3. Consider a class A:

```
class A {
public:
    string names; //data
    ..... //member functions as necessary
}
```

Suppose you want to build a Stack class consisting of A objects, with maximum prescribed capacity.

```
class Stack{
public:
    int top; //top of the stack, bottom is 0
    A data[MAX_CAPACITY]; //stack is modelled as an array of A objects
    bool push(A); //push onto the stack - return success or failure
    A pop(); //pop from stack - return the top of the stack
    Stack reverse(); //returns a new stack with elements in reverse order
    Stack(){
};
```

Write the necessary C++ code, and test it with a main() function.