Problem # 1
Print in Binary

Write a recursive function called **printBinary** to print an unsigned integer in binary. The prototype of the function should be like this:

void printBinary (unsigned int x)

No global memory or loops can be used in this task. (20% deduction will be applied if you cannot handle 0)

Sample Input(s)	**Corresponding Output(s)**
0 1 2 20 39	0 1 10 10100 100111

Problem # 2
Palindrome

Write a recursive function called **isPalindrome** to detect whether a string is palindrome or not. The prototype of the function should be like this:

int ispalindrome (char *str, int n)

where str contains the string in question and n represents its length. **No global memory or loops can be used in this task**.

Sample Input(s)	**Corresponding Output(s)**
madam	Yes
Mommy	No

Problem # 3
Rotate using Bitwise Operator

Write a function to n times right rotate an integer x. The prototype of the function should be like this:

int rightRotate (int x, int n)

Using this function write a program that takes two integers x and n, and returns x right rotated by n bit positions. Your program must output x in binary and then the rotated x in binary.

N.B.: Use the function implemented in Problem#1 to print x in binary.

Sample Input(s)	**Corresponding Output(s)**
20 2	10100 00101
53 3	110101 101110

Problem # 4
Even Parity

A bit pattern has even parity if the number of set bits in the pattern is even. (example: 1010, 010111 have even parity, but 1101 does not). Write a function that detects whether an integer has even parity or not. The prototype of the function should be like this:

bool parityCheck (int x)

Your program must output x in binary before calling the function.

Sample Input(s)	**Corresponding Output(s)**
11	1011 No
20	10100 Yes
44	101100 No
53	110101 Yes