Solutions to Coding Problems

# Problem 1: Finding the Winner of the Game

This problem can be solved using the Josephus problem concept. Here’s the C code for it:

#include <stdio.h>  
  
int findTheWinner(int n, int k) {  
 if (n == 1) return 1;  
 return (findTheWinner(n - 1, k) + k - 1) % n + 1;  
}  
  
int main() {  
 int n = 5, k = 2;  
 printf("The winner is: %d\n", findTheWinner(n, k));  
 return 0;  
}

# Problem 2: Roman Numeral to Integer Conversion

This problem involves converting a Roman numeral string to an integer. Here’s the C code for it:

#include <stdio.h>  
#include <string.h>  
  
int romanToInt(char \*s) {  
 int values[26] = {0};  
 values['I' - 'A'] = 1;  
 values['V' - 'A'] = 5;  
 values['X' - 'A'] = 10;  
 values['L' - 'A'] = 50;  
 values['C' - 'A'] = 100;  
 values['D' - 'A'] = 500;  
 values['M' - 'A'] = 1000;  
  
 int sum = 0;  
 for (int i = 0; i < strlen(s); i++) {  
 if (i + 1 < strlen(s) && values[s[i] - 'A'] < values[s[i + 1] - 'A']) {  
 sum -= values[s[i] - 'A'];  
 } else {  
 sum += values[s[i] - 'A'];  
 }  
 }  
 return sum;  
}  
  
int main() {  
 char s[] = "MCMXCIV";  
 printf("The integer value is: %d\n", romanToInt(s));  
 return 0;  
}

# Problem 3: Valid Parentheses

This problem can be solved using a stack to ensure matching brackets. Here’s the C code:

#include <stdio.h>  
#include <stdbool.h>  
#include <string.h>  
  
bool isValid(char \*s) {  
 char stack[10000];  
 int top = -1;  
  
 for (int i = 0; i < strlen(s); i++) {  
 char c = s[i];  
 if (c == '(' || c == '[' || c == '{') {  
 stack[++top] = c;  
 } else {  
 if (top == -1) return false;  
 char topChar = stack[top--];  
 if ((c == ')' && topChar != '(') ||  
 (c == ']' && topChar != '[') ||  
 (c == '}' && topChar != '{')) {  
 return false;  
 }  
 }  
 }  
 return top == -1;  
}  
  
int main() {  
 char s[] = "()[]{}";  
 printf("Is valid: %s\n", isValid(s) ? "true" : "false");  
 return 0;  
}

# Problem 4: Validating Phone Numbers

This problem can be solved with a one-liner bash script using regular expressions:

grep -E '^(\([0-9]{3}\) [0-9]{3}-[0-9]{4}|[0-9]{3}-[0-9]{3}-[0-9]{4})$' file.txt

# Problem 5: Printing the 10th Line

This problem can be solved using several methods in bash. Here are a few options:

## Method 1: Using sed

sed -n '10p' file.txt

## Method 2: Using awk

awk 'NR == 10' file.txt

## Method 3: Using head and tail

head -n 10 file.txt | tail -n 1