C Programs for Practice (Control Flow)

1. **C Program to check whether a number is positive, negative or zero**

```
// C Program to check if a number is positive, negative,
// or zero using simple conditional checks
#include <stdio.h>

void checkNum(int N) {

    // Check if the number is zero
    if (N == 0) {
        printf("Zeri\n");
    }
    // Check if the number is less than zero
    else if (N < 0) {
        printf("Negative\n");
    }
    // If neither, the number is positive
    else {
        printf("Positive\n");
    }
}

int main() {
    int N = 10;
    checkNum(N);
    return 0;
}
```

2. **C Program to check whether a number is even or odd**

```
// C Program to Check Even or Odd Using Modulo Operator
#include <stdio.h>

void checkOddEven(int N) {

    // Find the remainder
    int r = N % 2;

    // Condition for even
    if (r == 0)  {
        printf("Even");
    }
```

```
    // Condition for odd number
    else  {
        printf("Odd");
    }
}

int main() {
    int N = 101;
    checkOddEven(N);
    return 0;
}
```

3. C Program to check whether a character is vowel or constant

```
// C program to check if a character
// is a vowel or consonant
#include <stdio.h>

// Driver code
int main()
{
        char ch = 'A';

        // Checking if the character ch
        // is a vowel or not.
        if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E'
                || ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O'
                || ch == 'u' || ch == 'U') {

                printf("The character %c is a vowel.\n", ch);
        }
        else {
                printf("The character %c is a consonant.\n", ch);
        }

        return 0;
}
```

4. C Program to determine largest among three numbers

```
// C program to find the largest number among three number
// using nested if-else
#include <stdio.h>

int main()
{
   int c = 10, b = 22, a = 9;

   // Finding largest by comparing using relational operators
   if (a >= b) {
      if (a >= c)
         printf("%d is the largest number.", a);
      else
         printf("%d is the largest number.", c);
   }
   else {
      if (b >= c)
         printf("%d is the largest number.", b);
      else
         printf("%d is the largest number.", c);
   }

   return 0;
}
```

5. C Program to calculate sum of natural numbers

```
// C Program to demonstrate
// Sum of Natural Numbers
// using while loops

#include <stdio.h>
int main()
{
        int i, s = 0;
        int n = 10;
        i = 1;

        // while loop executes
        // the statements until the
        // condition is false
        while (i <= n) {
```

```
            // adding natural numbers
            // up to given number n
            s += i;
            i++;
        }
        // printing the result
        printf("Sum = %d", s);
        return 0;
}
```

6. C Program to print alphabets from A to Z using Loop

```
// C program to find the print
// Alphabets from A to Z

#include <stdio.h>

int main()
{
        // Declare the variables
        char i;

        // Display the alphabets
        printf("The Alphabets from A to Z are: \n");

        // Traverse each character
        // with the help of for loop
        for (i = 'A'; i <= 'Z'; i++) {

            // Print the alphabet
            printf("%c ", i);
        }

        printf("\nThe Alphabets from a to z are: \n");

        // Traverse each character
        // with the help of for loop
        for (i = 'a'; i <= 'z'; i++) {

            // Print the alphabet
            printf("%c ", i);
        }
```

```
	return 0;
}
```

7. C Program to check leap year

```
// C program to check if a given
// year is leap year or not
#include <stdbool.h>
#include <stdio.h>

bool checkYear(int year)
{
	// If a year is multiple of 400,
	// then it is a leap year
	if (year % 400 == 0)
		return true;

	// Else If a year is multiple of 100,
	// then it is not a leap year
	else if (year % 100 == 0)
		return false;

	// Else If a year is multiple of 4,
	// then it is a leap year
	else if (year % 4 == 0)
		return true;
	// if no above condition is satisfied, then it is not
	// a leap year
	return false;
}

// Driver code
int main()
{
	int year = 2000;

	if (checkYear(year)) {
		printf("Leap Year");
	}
	else {
		printf("Not a Leap Year");
	}
	return 0;
}
```

8. C Program to find the factorial of a number

```
// C program to Find the Factorial Using for Loop
#include <stdio.h>

unsigned int factorial(unsigned int N) {
    int fact = 1, i;

    // Loop from 1 to N to get the factorial
    for (i = 1; i <= N; i++) {
        fact *= i;
    }

    return fact;
}

int main() {
    int N = 5;
    int fact = factorial(N);
    printf("Factorial of %d is %d", N, fact);
    return 0;
}
```

9. C Program to print Fibonacci series

```
// C Program to print the fibonacci series using loops
#include <stdio.h>

void printFib(int n) {

      // If the number of terms is smaller than 1
    if (n < 1) {
        printf("Invalid Number of terms\n");
        return;
    }

      // First two terms of the series
    int prev1 = 1;
    int prev2 = 0;

    // for loop that prints n terms of fibonacci series
    for (int i = 1; i <= n; i++) {
```

```
        // Print current term and update previous terms
        if (i > 2) {
            int curr = prev1 + prev2;
            prev2 = prev1;
            prev1 = curr;
            printf("%d ", curr);
        }
        else if (i == 1)
            printf("%d ", prev2);
        else (i == 2)
            printf("%d ", prev1);
    }
}

int main() {
    int n = 9;

     // Printing first n fibonacci terms
    printFib(n);
    return 0;
}
```

10. C Program to find LCM of two numbers

```
// C program to find LCM of
// two numbers
#include <stdio.h>

// Driver code
int main()
{
    int x = 15, y = 25, max;
    max = (x > y) ? x : y;

    // While loop to check if max variable
    // is divisible by x and y
    while (1) {
        if (max % x == 0 && max % y == 0) {
            printf("The LCM of %d and %d is %d.", x, y,
                   max);
            break;
        }

        ++max;
```

```
    }

    return 0;
}
```

11. C Program to check Armstrong number

```
// C program to check given number is Armstrong number
// or not using Functions
#include <math.h>
#include <stdio.h>
#include <stdbool.h>

bool isArmstrong(int N) {
    int temp = N;
    int sum = 0;

    // Get the number of digits
      // Adding 1 to compensate for the loss of fraction part
      // of the value returned by log10 due to the conversion
    // into integer
    int K = log10(temp) + 1;

    // Calculate the sum of the digits raised to the power of
    // num_digits
    while (temp > 0) {
        int digit = temp % 10;
        sum += pow(digit, K);
        temp /= 10;
    }

    // Return whether the sum is equal to the original number or not
    return (sum == N);
}

int main() {
    int N = 153;

    // Check if the number is an Armstrong number
    if (isArmstrong(N)) {
        printf("Yes\n");
    }
    else {
        printf("No\n");
```

```
    }

    return 0;
}
```

12. C Program to reverse a number

```
// C program to implement
// reverse a number
#include <stdio.h>

// Iterative function to
// reverse digits of num
int reverseDigits(int num)
{
        int rev_num = 0;
        while (num > 0) {
                rev_num = rev_num * 10 + num % 10;
                num = num / 10;
        }
        return rev_num;
}

// Driver code
int main()
{
        int num = 4562;
        printf("Reverse of is %d", reverseDigits(num));

        getchar();
        return 0;
}
```

13. C Program to check for Palindrome number

```
// C Program to check if a number is a palindrome by
// reversing the number
#include <stdio.h>

int reverseNum(int N) {

    // Function to store the reversed number
    int rev = 0;
    while (N > 0) {
```

```
        // Extract the last digit
        int dig = N % 10;

        // Append the digit to the reversed number
        rev = rev * 10 + dig;

        // Remove the last digit
        N /= 10;
    }
    return rev;
}

int isPalindrome(int N) {

    // Negative numbers are not palindromes
    if (N < 0)
        return 0;
    return N == reverseNum(N);
}

int main() {
    int N = 121;
    if (isPalindrome(N)) {
        printf("Yes\n");
    }
    else {
        printf("No\n");
    }
    return 0;
}
```

14. C Program to check for Prime number

```
// C Program to check for prime number using
//  Simple Trial Division
#include <stdbool.h>
#include <stdio.h>

int main() {
    int n = 29;

    int cnt = 0;
```

```
    // If number is less than/equal to 1,
    // it is not prime
    if (n <= 1)
        printf("%d is NOT prime\n", n);
    else {

        // Check for divisors from 1 to n
        for (int i = 1; i <= n; i++) {

            // Check how many number is divisible
            // by n
            if (n % i == 0)
                cnt++;
        }

        // If n is divisible by more than 2 numbers
        // then it is not prime
        if (cnt > 2)
            printf("%d is NOT prime\n", n);

        // else it is prime
        else
            printf("%d is prime", n);
    }

    return 0;
}
```

15. C Program to find all factors of a Natural Number

```
// C implementation of Naive
// method to print all divisors
#include <stdio.h>

// Function to print the divisors
void printDivisors(int n)
{
        for (int i = 1; i <= n; i++)
                if (n % i == 0)
                        printf("%d ", i);
}

// Driver code
int main()
```

```
{
        printf("The divisors of 100 are: ");
        printDivisors(100);

        return 0;
}
```

16. C Program to find the HCF/GCD of two numbers

```
// C program to find GCD of two numbers

#include <math.h>
#include <stdio.h>

// Function to return gcd of a and b
int gcd(int a, int b)
{
   // Find Minimum of a and b
   int result = ((a < b) ? a : b);
   while (result > 0) {
      if (a % result == 0 && b % result == 0) {
         break;
      }
      result--;
   }

   // Return gcd of a and b
   return result;
}

// Driver code
int main()
{
   int a = 98, b = 56;
   printf("GCD of %d and %d is %d ", a, b, gcd(a, b));
   return 0;
}
```