C programming-Day 3

Loops and Nested Loops

1. Write a C Program to print the multiplication table of a given number using a for loop.

```
#include <stdio.h>
int main() {
int num, i;
printf("Enter a number: ");
scanf("%d", &num);
printf("Multiplication table of %d:\n", num);
for (i = 1; i <= 10; ++i) {
printf("%d x %d = %d\n", num, i, num * i);
}
return 0;
}
```

2. Write a C Program to find the sum of all even numbers from 1 to n using a while loop.

```
#include <stdio.h>
int main() {
int n, i = 2, sum = 0;
printf("Enter a number: ");
scanf("%d", &n);
while (i <= n) {
sum += i;
i += 2;
}
printf("Sum of all even numbers from 1 to %d is: %d\n", n, sum);
return 0;
}
```

3. Write a C Program to check whether a number is a palindrome using a while loop.

```
#include <stdio.h>
int main() {
int num, originalNum, reversedNum = 0, remainder;
printf("Enter a number: ");
scanf("%d", &num);
originalNum = num;
while (num != 0) {
```

```
  remainder = num % 10;
  reversedNum = reversedNum * 10 + remainder;
  num /= 10;
 }
 if (originalNum == reversedNum)
 printf("%d is a palindrome.\n", originalNum);
 else
 printf("%d is not a palindrome.\n", originalNum);
 return 0;
}
```

4. Write a C Program to display the following pattern using nested loops:

```
*
**
***
****
*****
```

```
#include <stdio.h>
int main() {
int i, j;
for (i = 1; i <= 5; i++)
 {
    for (j = 1; j <= i; j++)
    {    printf("*");   }
      printf("\n");
 }
return 0;
}
```

5. Write a C Program to display all prime numbers between two given numbers using nested loops.

```
#include <stdio.h>
int main() {
int start, end, i, j, isprime;
printf("Enter the starting number: ");
scanf("%d", &start);
printf("Enter the ending number: ");
scanf("%d", &end);
printf("Prime numbers between %d and %d are:\n", start, end);
```

```
for (i = start; i <= end; i++) {
if (i < 2) continue;
isprime = 1;
for (j = 2; j <= i / 2; j++) {
if (i % j == 0) {
isprime = 0;
break;
}
}
if (isPrime) {
printf("%d ", i);
}
}
printf("\n");
return 0;
}
```

6. Write a C Program to find the factorial of a number using a for loop.

```
#include <stdio.h>
int main() {
int num, i;
double factorial = 1;
printf("Enter a number to find its factorial: ");
scanf("%d", &num);

if (num < 0)
 {
printf("Factorial of negative numbers doesn't exist.\n");
}
else {
for (i = 1; i <= num; i++) {
factorial *= i;  }
printf("Factorial of %d is: %0.3lf\n", num, factorial);
}
return 0;
}
```

7. Write a C Program to print the following number pattern using nested loops:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

```
#include <stdio.h>
int main() {
int i, j;
for (i = 1; i <= 5; i++)
{
for (j = 1; j <= i; j++)

printf("%d ", j);
}
printf("\n");
}
return 0;
}
```

8. Write a C Program to reverse the digits of a given number using a while loop.

```
#include <stdio.h>
int main() {
int num, reversed = 0, remainder;
printf("Enter a number to reverse: ");
scanf("%d", &num);
while (num != 0)
{
remainder = num % 10;
reversed = reversed * 10 + remainder;
num /= 10;
}
printf("Reversed number is: %d\n", reversed);
return 0;
}
```

Functions

1. Write a C Program to create a function to calculate the square of a number.

```
#include <stdio.h>
int square(int num) {
return num * num;
}

int main() {
int number;
printf("Enter a number to calculate its square: ");
scanf("%d", &number);
printf("Square of %d is: %d\n", number, square(number));
return 0;
}
```

2. Write a C Program to create a function to check whether a number is prime or not.

```
#include <stdio.h>
int isPrime(int num) {
if (num < 2) return 0;
for (int i = 2; i <= num / 2; i++) {
if (num % i == 0) return 0;     }
return 1; // Prime
}

int main() {
int number;
printf("Enter a number to check if it's prime: ");
scanf("%d", &number);
if (isPrime(number)) {
printf("%d is a prime number.\n", number);
} else {
printf("%d is not a prime number.\n", number);
}
return 0;
}
```

3. Write a C Program to create a function to calculate the sum of digits of a number.

```
#include <stdio.h>
int sumOfDigits(int num) {
int sum = 0;
while (num != 0) {
sum += num % 10; num /= 10; }
return sum;
}
int main() {
int number;
printf("Enter a number to calculate the sum of its digits: ");
scanf("%d", &number);
printf("Sum of digits of %d is: %d\n", number, sumOfDigits(number));
return 0;
}
```

4. Write a C Program to create a function to swap two numbers using call by reference.

```
#include <stdio.h>
void swap(int *a, int *b) {
int temp = *a;
*a = *b;
*b = temp;
}
int main() {
int num1, num2;
printf("Enter two numbers to swap:\n");
scanf("%d %d", &num1, &num2);
printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
swap(&num1, &num2);
printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
return 0;
}
```

5. Write a C Program to create a function to find the maximum of three numbers using if-else.

```
#include <stdio.h>
int max(int a, int b, int c) {
if (a >= b && a >= c)
return a;
else if (b >= a && b >= c)
return b;
else
return c;
}
int main() {
int num1, num2, num3;
printf("Enter three numbers: ");
scanf("%d %d %d", &num1, &num2, &num3);
printf("The maximum of %d, %d, and %d is: %d\n", num1, num2, num3, max(num1, num2, num3));
return 0;
}
```

6. Write a C Program to create a function to check if a string is a palindrome using recursion.

```
#include <stdio.h>
#include <string.h>
int palindrome(char str[], int start, int end) {
if (start >= end)
return 1;
if (str[start] != str[end])
return 0;
return isPalindrome(str, start + 1, end - 1);
}

int main() {
char str[100];
printf("Enter a string: ");
scanf("%s", str);

if (isPalindrome(str, 0, strlen(str) - 1)) {
printf("The string is a palindrome.\n");
}
```

```
else {
printf("The string is not a palindrome.\n");
}
return 0;
}
```

7. Write a C Program to create a function to generate the first n Fibonacci numbers using recursion.

```
#include <stdio.h>
int fibonacci(int n) {
if (n <= 1)
return n;
return fibonacci(n - 1) + fibonacci(n - 2);
}
int main() {
int n;
printf("Enter the number of Fibonacci terms: ");
scanf("%d", &n);
printf("The first %d Fibonacci numbers are:\n", n);
for (int i = 0; i < n; i++) {
printf("%d ", fibonacci(i));
}
printf("\n");
return 0;
}
```

8 Write a C Program to create a function to calculate the power of a number (x^y) using recursion.

```
#include <stdio.h>
int power(int x, int y) {
if (y == 0)
return 1;
return x * power(x, y - 1);
}
int main() {
int x, y;
printf("Enter the base (x): ");
scanf("%d", &x);
printf("Enter the exponent (y): ");
```

```
scanf("%d", &y);
printf("%d^%d = %d\n", x, y, power(x, y));
return 0;
}
```