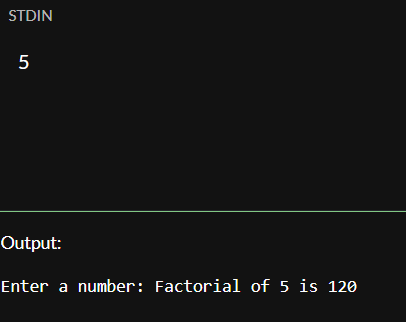
Day : Functions (8-8-2025)

# 1. Write a function to find the factorial of a number.

IPO (Input - Process - Output):

Input: An integer n  
Process: Multiply 1 to n using recursion  
Output: Factorial of the number

Code:

#include <stdio.h>  
int factorial(int n) {  
 if (n == 0 || n == 1) return 1;  
 return n \* factorial(n - 1);  
}  
int main() {  
 int num;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 printf("Factorial of %d is %d", num, factorial(num));  
 return 0;  
}  
 

Output (Example):

Enter a number: 5  
Factorial of 5 is 120

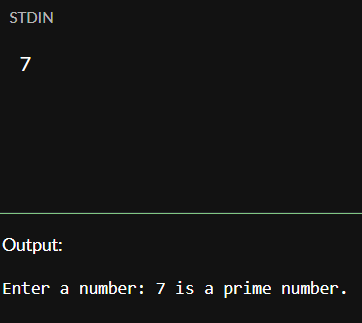
# 2. Write a function to check whether a number is prime.

IPO (Input - Process - Output):

Input: An integer  
Process: Check divisibility from 2 to n/2  
Output: Prime or not

Code:

#include <stdio.h>  
int isPrime(int n) {  
 if (n <= 1) return 0;  
 for (int i = 2; i <= n / 2; i++) {  
 if (n % i == 0) return 0;  
 }  
 return 1;  
}  
int main() {  
 int num;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 if (isPrime(num))  
 printf("%d is a prime number.", num);  
 else  
 printf("%d is not a prime number.", num);  
 return 0;  
}



Output (Example):

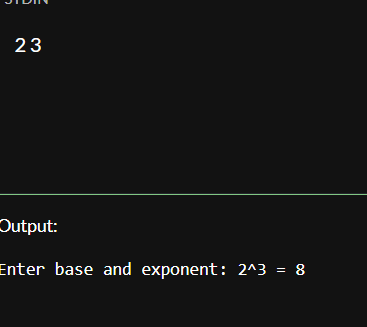
Enter a number: 7  
7 is a prime number.

# 3. Write a function to calculate power using recursion.

IPO (Input - Process - Output):

Input: Base and exponent  
Process: Multiply base recursively  
Output: Power result

Code:

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

Output (Example):

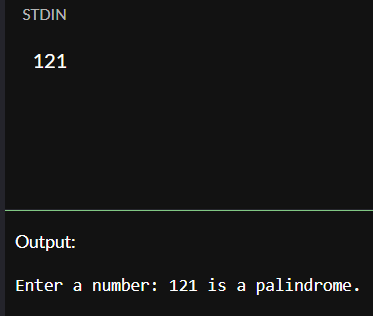
Enter base and exponent: 2 3  
2^3 = 8

# 4. Write a function to check palindrome number using recursion.

IPO (Input - Process - Output):

Input: A number  
Process: Reverse the number using recursion and compare  
Output: Palindrome or not

Code:

#include <stdio.h>  
int reverse(int n, int rev) {  
 if (n == 0) return rev;  
 return reverse(n / 10, rev \* 10 + n % 10);  
}  
int main() {  
 int num, revNum;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 revNum = reverse(num, 0);  
 if (num == revNum)  
 printf("%d is a palindrome.", num);  
 else  
 printf("%d is not a palindrome.", num);  
 return 0;  
}  
 

Output (Example):

Enter a number: 121  
121 is a palindrome.

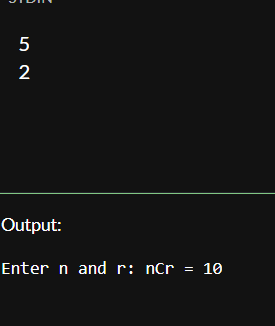
# 5. Write a function to calculate nCr (combinations).

IPO (Input - Process - Output):

Input: Two integers n and r  
Process: Use formula nCr = n! / (r!(n-r)!)  
Output: Value of nCr

Code:

#include <stdio.h>  
int fact(int n) {  
 if (n == 0) return 1;  
 return n \* fact(n - 1);  
}  
int combination(int n, int r) {  
 return fact(n) / (fact(r) \* fact(n - r));  
}  
int main() {  
 int n, r;  
 printf("Enter n and r: ");  
 scanf("%d%d", &n, &r);  
 printf("nCr = %d", combination(n, r));  
 return 0;  
}



Output (Example):

Enter n and r: 5 2  
nCr = 10

# 6. Write a program to demonstrate call by value and call by reference.

IPO (Input - Process - Output):

Input: Two integers  
Process: Show difference between value and reference  
Output: Swapped values for reference

Code:

#include <stdio.h>  
void swapByValue(int a, int b) {  
 int temp = a;  
 a = b;  
 b = temp;  
 printf("Inside swapByValue: a = %d, b = %d\n", a, b);  
}  
void swapByReference(int \*a, int \*b) {  
 int temp = \*a;  
 \*a = \*b;  
 \*b = temp;  
}  
int main() {  
 int x = 10, y = 20;  
 swapByValue(x, y);  
 printf("After call by value: x = %d, y = %d\n", x, y);  
 swapByReference(&x, &y);  
 printf("After call by reference: x = %d, y = %d", x, y);  
 return 0;  
}

Output (Example):

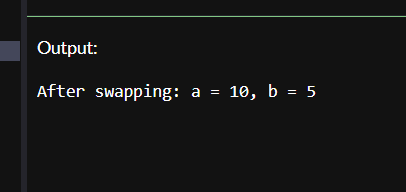
Inside swapByValue: a = 20, b = 10  
After call by value: x = 10, y = 20  
After call by reference: x = 20, y = 10

# 7. Write a program using function to swap two numbers.

IPO (Input - Process - Output):

Input: Two integers  
Process: Swap using temporary variable  
Output: Swapped values

Code:

#include <stdio.h>  
void swap(int \*x, int \*y) {  
 int temp = \*x;  
 \*x = \*y;  
 \*y = temp;  
}  
int main() {  
 int a = 5, b = 10;  
 swap(&a, &b);  
 printf("After swapping: a = %d, b = %d", a, b);  
 return 0;  
}  
 

Output (Example):

After swapping: a = 10, b = 5

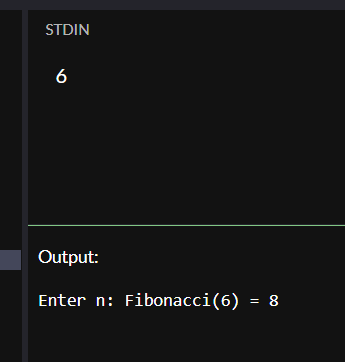
# 8. Write a recursive function to find the nth Fibonacci number.

IPO (Input - Process - Output):

Input: An integer n  
Process: Recursively calculate nth Fibonacci  
Output: nth Fibonacci number

Code:

#include <stdio.h>  
int fibonacci(int n) {  
 if (n == 0) return 0;  
 if (n == 1) return 1;  
 return fibonacci(n - 1) + fibonacci(n - 2);  
}  
int main() {  
 int n;  
 printf("Enter n: ");  
 scanf("%d", &n);  
 printf("Fibonacci(%d) = %d", n, fibonacci(n));  
 return 0;  
}



Output (Example):

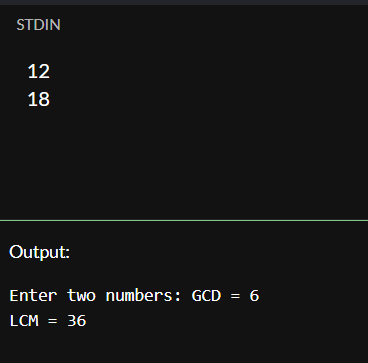
Enter n: 6  
Fibonacci(6) = 8

# 9. Write a program to find GCD and LCM using functions.

IPO (Input - Process - Output):

Input: Two integers  
Process: GCD using Euclidean algorithm, LCM = a\*b/GCD  
Output: GCD and LCM

Code:

#include <stdio.h>  
int gcd(int a, int b) {  
 while (b != 0) {  
 int temp = b;  
 b = a % b;  
 a = temp;  
 }  
 return a;  
}  
int lcm(int a, int b) {  
 return (a \* b) / gcd(a, b);  
}  
int main() {  
 int a, b;  
 printf("Enter two numbers: ");  
 scanf("%d%d", &a, &b);  
 printf("GCD = %d\nLCM = %d", gcd(a, b), lcm(a, b));  
 return 0;  
}  
 

Output (Example):

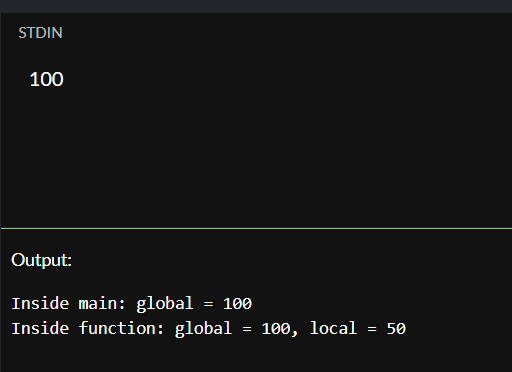
Enter two numbers: 12 18  
GCD = 6  
LCM = 36

# 10. Write a program to demonstrate global and local variables.

IPO (Input - Process - Output):

Input: None  
Process: Show scope of global and local variables  
Output: Printed values of each

Code:

#include <stdio.h>  
int globalVar = 100;  
void display() {  
 int localVar = 50;  
 printf("Inside function: global = %d, local = %d\n", globalVar, localVar);  
}  
int main() {  
 printf("Inside main: global = %d\n", globalVar);  
 display();  
 return 0;  
}  
 Output (Example):

Inside main: global = 100  
Inside function: global = 100, local = 50