**Assignment(8/8/2025)**

### ✅ 1. **Factorial of a Number**

**IPO:**

* Input: Integer n
* Process: Use recursion to compute n!
* Output: Factorial of n

**Program:**

#include <stdio.h>

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

int main() {

int n = 5;

printf("Factorial of %d is %d\n", n, factorial(n));

return 0;

}

**Sample Output:**

Factorial of 5 is 120

### ✅ 2. **Check Prime Number**

**IPO:**

* Input: Integer n
* Process: Check if divisible by numbers < n
* Output: Whether n is prime or not

**Program:**

#include <stdio.h>

int isPrime(int n) {

if (n <= 1) return 0;

for(int i = 2; i < n; i++) {

if(n % i == 0) return 0;

}

return 1;

}

int main() {

int n = 7;

if (isPrime(n))

printf("%d is a Prime number\n", n);

else

printf("%d is not a Prime number\n", n);

return 0;

}

**Sample Output:**

7 is a Prime number

### ✅ 3. **Power using Recursion**

**IPO:**

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

**Program:**

#include <stdio.h>

int power(int base, int exp) {

if (exp == 0) return 1;

return base \* power(base, exp - 1);

}

int main() {

int base = 2, exp = 3;

printf("%d^%d = %d\n", base, exp, power(base, exp));

return 0;

}

**Sample Output:**

2^3 = 8

### ✅ 4. **Palindrome using Recursion**

**IPO:**

* Input: Number
* Process: Reverse using recursion and compare
* Output: Whether number is palindrome

**Program:**

#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 = 121;

int rev = reverse(num, 0);

if(num == rev)

printf("%d is a Palindrome\n", num);

else

printf("%d is not a Palindrome\n", num);

return 0;

}

**Sample Output:**

121 is a Palindrome

### ✅ 5. **nCr (Combinations)**

**IPO:**

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

**Program:**

#include <stdio.h>

int factorial(int n) {

if(n == 0) return 1;

return n \* factorial(n - 1);

}

int nCr(int n, int r) {

return factorial(n) / (factorial(r) \* factorial(n - r));

}

int main() {

int n = 5, r = 2;

printf("nCr of %dC%d = %d\n", n, r, nCr(n, r));

return 0;

}

**Sample Output:**

nCr of 5C2 = 10

### ✅ 6. **Call by Value & Reference**

**IPO:**

* Input: Two numbers
* Process: Modify using value and pointer
* Output: Show changes

**Program:**

#include <stdio.h>

void callByValue(int a) {

a = a + 10;

printf("Inside callByValue: %d\n", a);

}

void callByReference(int \*a) {

\*a = \*a + 10;

printf("Inside callByReference: %d\n", \*a);

}

int main() {

int x = 5;

callByValue(x);

printf("After callByValue: %d\n", x);

callByReference(&x);

printf("After callByReference: %d\n", x);

return 0;

}

**Sample Output:**

Inside callByValue: 15

After callByValue: 5

Inside callByReference: 15

After callByReference: 15

### ✅ 7. **Swap Two Numbers using Function**

**IPO:**

* Input: Two numbers
* Process: Swap using pointer
* Output: Swapped values

**Program:**

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x = 10, y = 20;

swap(&x, &y);

printf("After swapping: x = %d, y = %d\n", x, y);

return 0;

}

**Sample Output:**

After swapping: x = 20, y = 10

### ✅ 8. **Fibonacci using Recursion**

**IPO:**

* Input: Integer n
* Process: Return nth Fibonacci recursively
* Output: nth Fibonacci number

**Program:**

#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 = 6;

printf("Fibonacci(%d) = %d\n", n, fibonacci(n));

return 0;

}

**Sample Output:**

Fibonacci(6) = 8

### ✅ 9. **GCD and LCM using Function**

**IPO:**

* Input: Two integers
* Process: GCD using loop; LCM using formula
* Output: GCD and LCM

**Program:**

#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 = 12, b = 18;

printf("GCD = %d\n", gcd(a, b));

printf("LCM = %d\n", lcm(a, b));

return 0;

}

**Sample Output:**

GCD = 6

LCM = 36

### ✅ 10. **Global and Local Variables**

**IPO:**

* Input: No input needed
* Process: Show global vs local
* Output: Values of variables

**Program:**

#include <stdio.h>

int globalVar = 10;

void display() {

int localVar = 5;

printf("Global: %d, Local: %d\n", globalVar, localVar);

}

int main() {

display();

globalVar = 20;

display();

return 0;

}

**Sample Output:**

Global: 10, Local: 5

Global: 20, Local: 5

**Thank you**