

EXPERIMENT6: FUNCTIONS

Experiment 1: Factorial (Recursive & Non-Recursive) + Binomial Coefficient

Aim

To develop recursive and non-recursive functions to compute factorial of a number and use them to calculate the binomial coefficient for different values of n and r .

Algorithm

Factorial (Recursive)

1. If $n = 0$, return 1
2. Else return $n \times \text{FACT}(n-1)$

Factorial (Non-Recursive)

1. Set $\text{fact} = 1$
2. Run loop $i = 1$ to n
3. $\text{fact} = \text{fact} \times i$
4. Return fact

Binomial Coefficient

1. Accept n, r
2. Use formula:

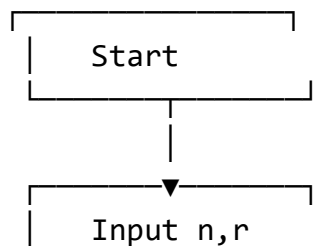
```
[
nCr = \frac{n!}{r!(n-r)!}
]
```

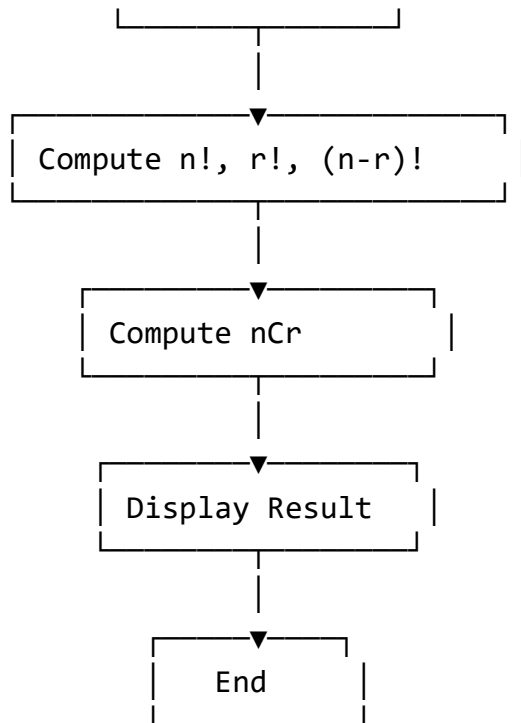
3. Display result

Pseudocode

```
FUNCTION factorial_rec(n)
  IF n == 0 THEN
    RETURN 1
  ELSE
    RETURN n * factorial_rec(n-1)
END FUNCTION
FUNCTION factorial_nonrec(n)
  fact = 1
  FOR i = 1 TO n
    fact = fact * i
  END FOR
  RETURN fact
END FUNCTION
READ n, r
nr = factorial(n) / (factorial(r) * factorial(n-r))
DISPLAY nr
```

Flowchart (ASCII)





C Program

```
#include <stdio.h>

// Recursive factorial
int fact_rec(int n) {
    if (n == 0) return 1;
    return n * fact_rec(n - 1);
}

// Non-recursive factorial
int fact_nonrec(int n) {
    int fact = 1;
    for (int i = 1; i <= n; i++)
        fact *= i;
    return fact;
}
```

```

int main() {
    int n, r;

    printf("Enter n and r: ");
    scanf("%d %d", &n, &r);

    int fr = fact_rec(n);
    int fn = fact_nonrec(n);

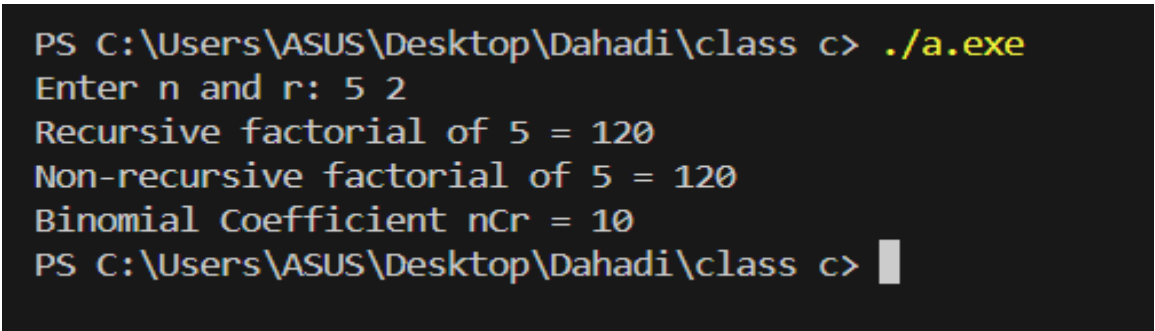
    int nCr = fact_rec(n) / (fact_rec(r) * fact_rec(n - r));

    printf("Recursive factorial of %d = %d\n", n, fr);
    printf("Non-recursive factorial of %d = %d\n", n, fn);
    printf("Binomial Coefficient nCr = %d\n", nCr);

    return 0;
}

```

Output



```

PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
Enter n and r: 5 2
Recursive factorial of 5 = 120
Non-recursive factorial of 5 = 120
Binomial Coefficient nCr = 10
PS C:\Users\ASUS\Desktop\Dahadi\class c>

```

Conclusion

Both recursive and non-recursive functions correctly compute factorial. Using factorial values, the binomial coefficient was successfully calculated.

Experiment 2: Recursive GCD Function

Aim

To develop a recursive function to find the GCD of two integers.

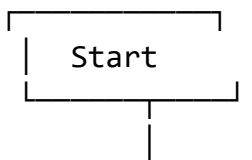
Algorithm

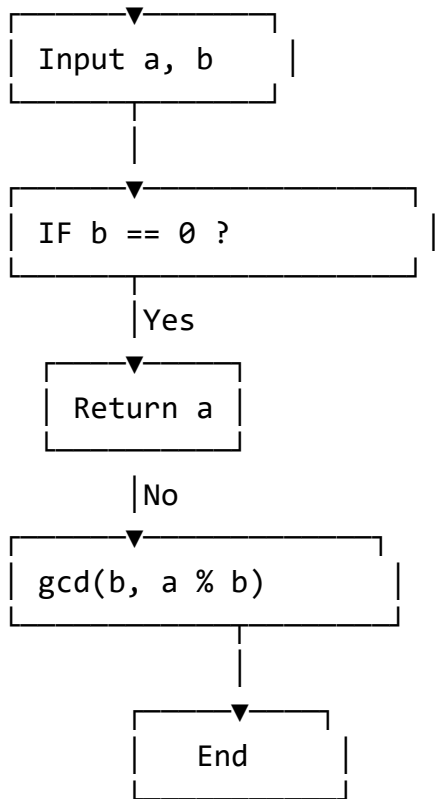
1. If $b = 0$ return a
2. Else return $\text{GCD}(b, a \% b)$

Pseudocode

```
FUNCTION gcd(a, b)
    IF b == 0 THEN
        RETURN a
    ELSE
        RETURN gcd(b, a % b)
END FUNCTION
```

Flowchart (ASCII)





C Program

```
#include <stdio.h>

int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}

int main() {
    int a, b;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);

    printf("GCD = %d\n", gcd(a, b));
    return 0;
}
```

```
}
```

Output

```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc fun1.c
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
Enter two integers: 6
8
GCD = 2
PS C:\Users\ASUS\Desktop\Dahadi\class c> █
```

Conclusion

The recursive function successfully computes the greatest common divisor using Euclid's algorithm.

Experiment 3: Recursive Fibonacci Function

Aim

To develop a recursive function to generate the Fibonacci series up to n.

Algorithm

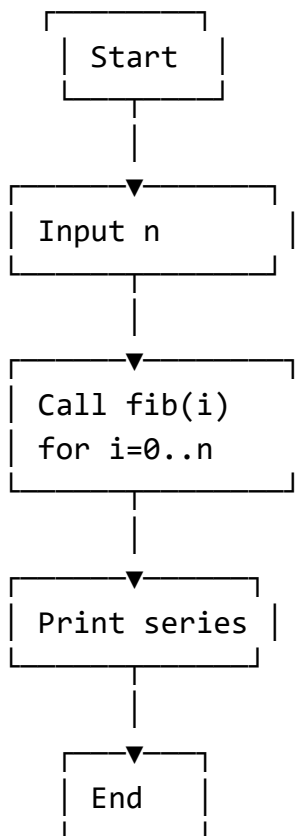
1. If $n == 0$ return 0
2. If $n == 1$ return 1

3. Else return $F(n-1) + F(n-2)$

Pseudocode

```
FUNCTION fib(n)
  IF n == 0 RETURN 0
  IF n == 1 RETURN 1
  RETURN fib(n-1) + fib(n-2)
END FUNCTION
```

Flowchart (ASCII)



C Program

```
#include <stdio.h>

int fib(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;
    return fib(n - 1) + fib(n - 2);
}

int main() {
    int n;
    printf("Enter limit: ");
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
        printf("%d ", fib(i));

    return 0;
}
```

Output

```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc fun1.c
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
Enter limit: 4
0 1 1 2
PS C:\Users\ASUS\Desktop\Dahadi\class c> |
```

Conclusion

The Fibonacci series was successfully generated using recursive function calls.

Experiment 4: Prime Number Function

Aim

To develop a function ISPRIME(num) to test if a number is prime and print all primes in a given range.

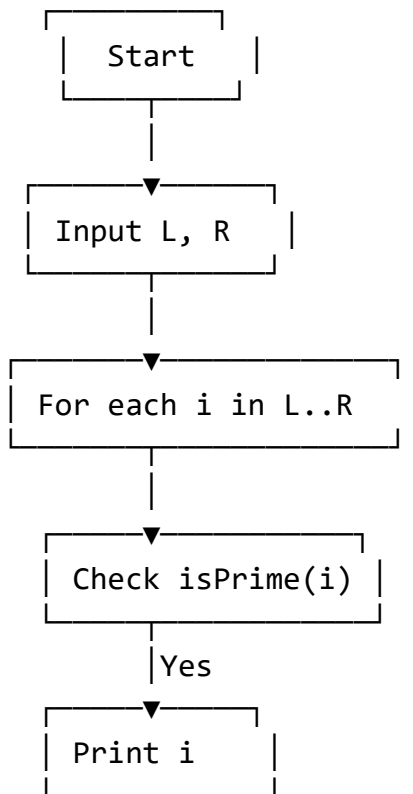
Algorithm

1. If $n < 2$ return false
2. Check divisibility from 2 to \sqrt{n}
3. If divisible \rightarrow not prime
4. Else prime

Pseudocode

```
FUNCTION isPrime(n)
    IF n < 2 RETURN 0
    FOR i = 2 TO sqrt(n)
        IF n % i == 0 RETURN 0
    RETURN 1
END FUNCTION
```

Flowchart (ASCII)



C Program

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

int isPrime(int n) {
    if (n < 2) return 0;
    for (int i = 2; i <= sqrt(n); i++)
        if (n % i == 0) return 0;
    return 1;
}

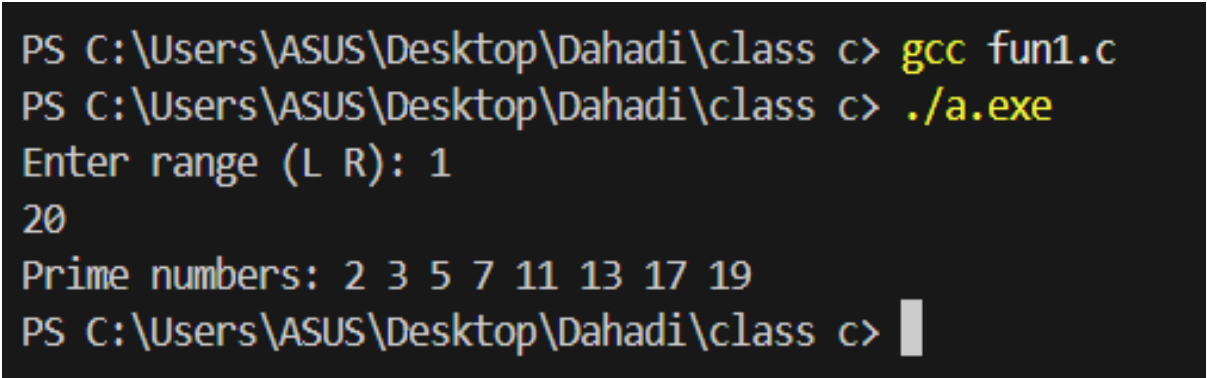
int main() {
    int L, R;
```

```
printf("Enter range (L R): ");
scanf("%d %d", &L, &R);

printf("Prime numbers: ");
for (int i = L; i <= R; i++)
    if (isPrime(i))
        printf("%d ", i);

return 0;
}
```

Output



```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc fun1.c
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe
Enter range (L R): 1
20
Prime numbers: 2 3 5 7 11 13 17 19
PS C:\Users\ASUS\Desktop\Dahadi\class c> |
```

Conclusion

The program successfully identifies prime numbers within the specified range.

Experiment 5: Reverse a String

Aim

To develop a function that reverses a string using character processing.

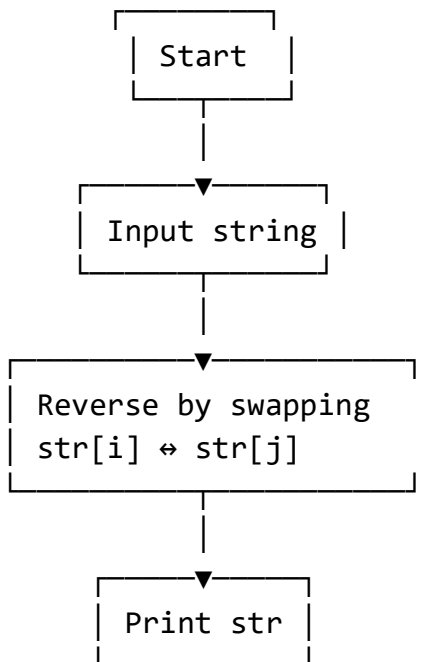
Algorithm

1. Read input string
2. Find length
3. Swap characters from both ends
4. Print reversed string

Pseudocode

```
FUNCTION reverse(str)
    i = 0
    j = length(str) - 1
    WHILE i < j
        swap str[i] and str[j]
        i++
        j--
    END FUNCTION
```

Flowchart (ASCII)



C Program

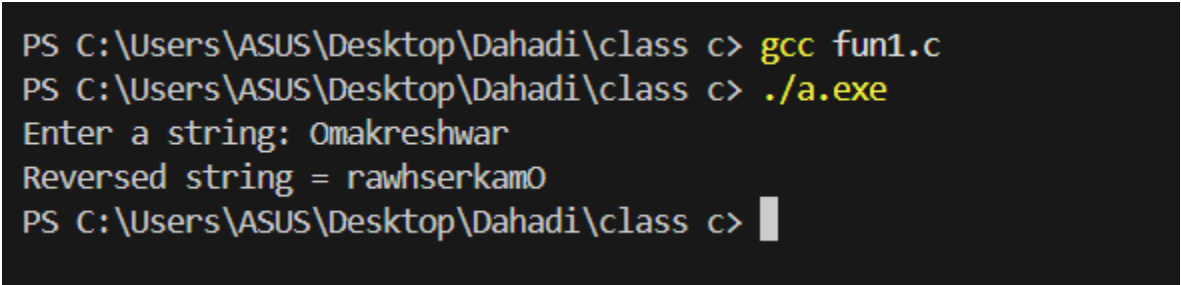
```
#include <stdio.h>
#include <string.h>

void reverse(char str[]) {
    int i = 0;
    int j = strlen(str) - 1;

    while (i < j) {
        char temp = str[i];
        str[i] = str[j];
        str[j] = temp;
        i++;
        j--;
    }
}
```

```
int main() {  
    char str[50];  
  
    printf("Enter a string: ");  
    scanf("%s", str);  
  
    reverse(str);  
  
    printf("Reversed string = %s\n", str);  
  
    return 0;  
}
```

Output



```
PS C:\Users\ASUS\Desktop\Dahadi\class c> gcc fun1.c  
PS C:\Users\ASUS\Desktop\Dahadi\class c> ./a.exe  
Enter a string: Omakreshwar  
Reversed string = rawhserkamO  
PS C:\Users\ASUS\Desktop\Dahadi\class c> █
```

Conclusion

The reverse() function works correctly by swapping characters from both ends.