

Write algorithms for basic problems such as finding the largest of three numbers, calculating the factorial of a number, and generating Fibonacci series. Draw flowcharts for the above algorithms using standard symbols.

**Algorithm for factorial of a given number.**

**Step 1:** Start.

**Step 2:** Read n.

**Step 3:** Initialize i=1,fact=1.

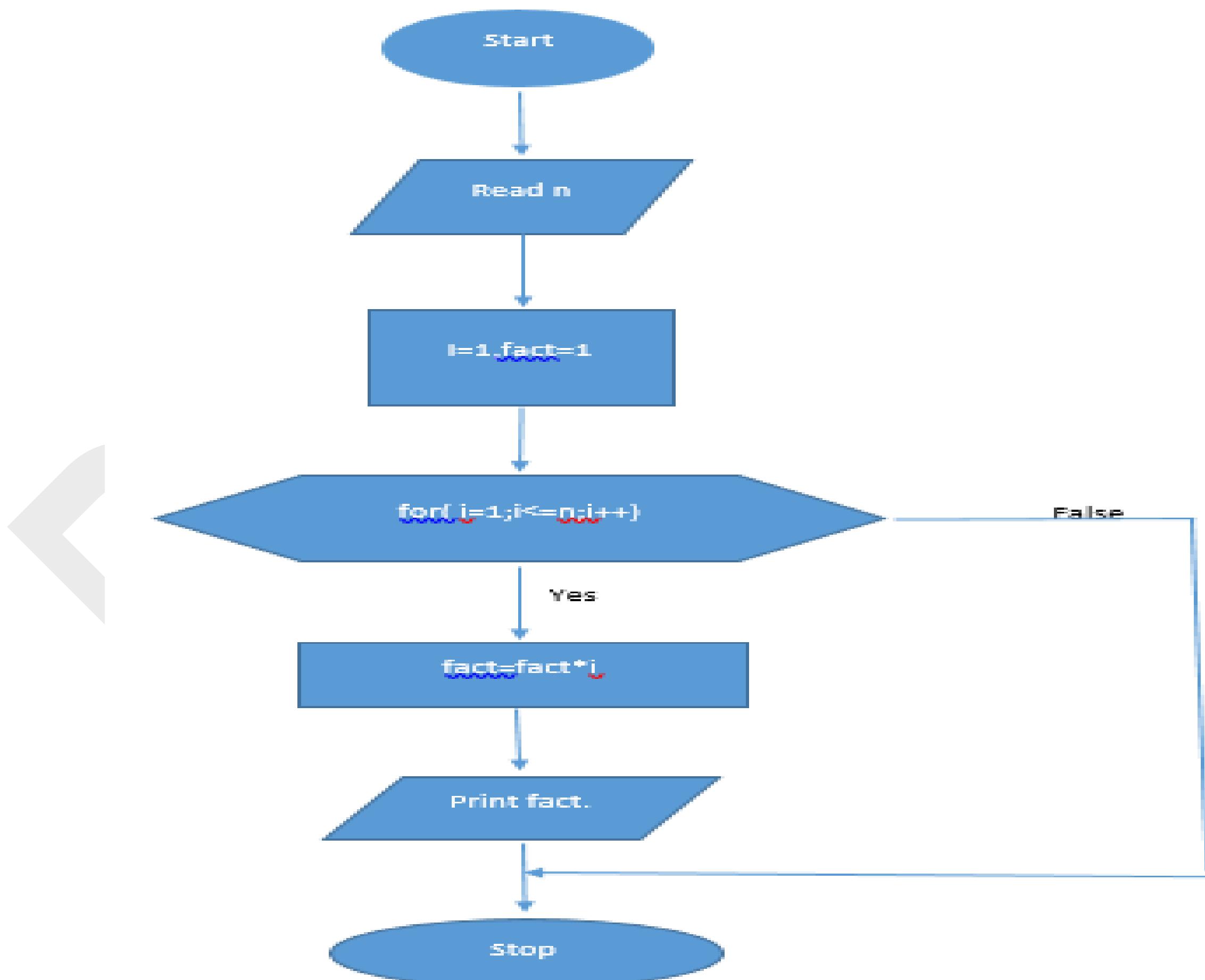
**Step4:** Repeat this step until i<=n.

**Step5:** fact=fact\*i.

**Step6:** Print fact.

**Step7:** Stop.

**Flowchart for factorial of a given number.**



**Algorithm for largest of three numbers.****Step 1:** Start.**Step 2:** Read a,b,c.**Step 3:** if( $a > b$ ) Then    if( $a > c$ ) Then

large=a.

else

large=c.

end if.

else

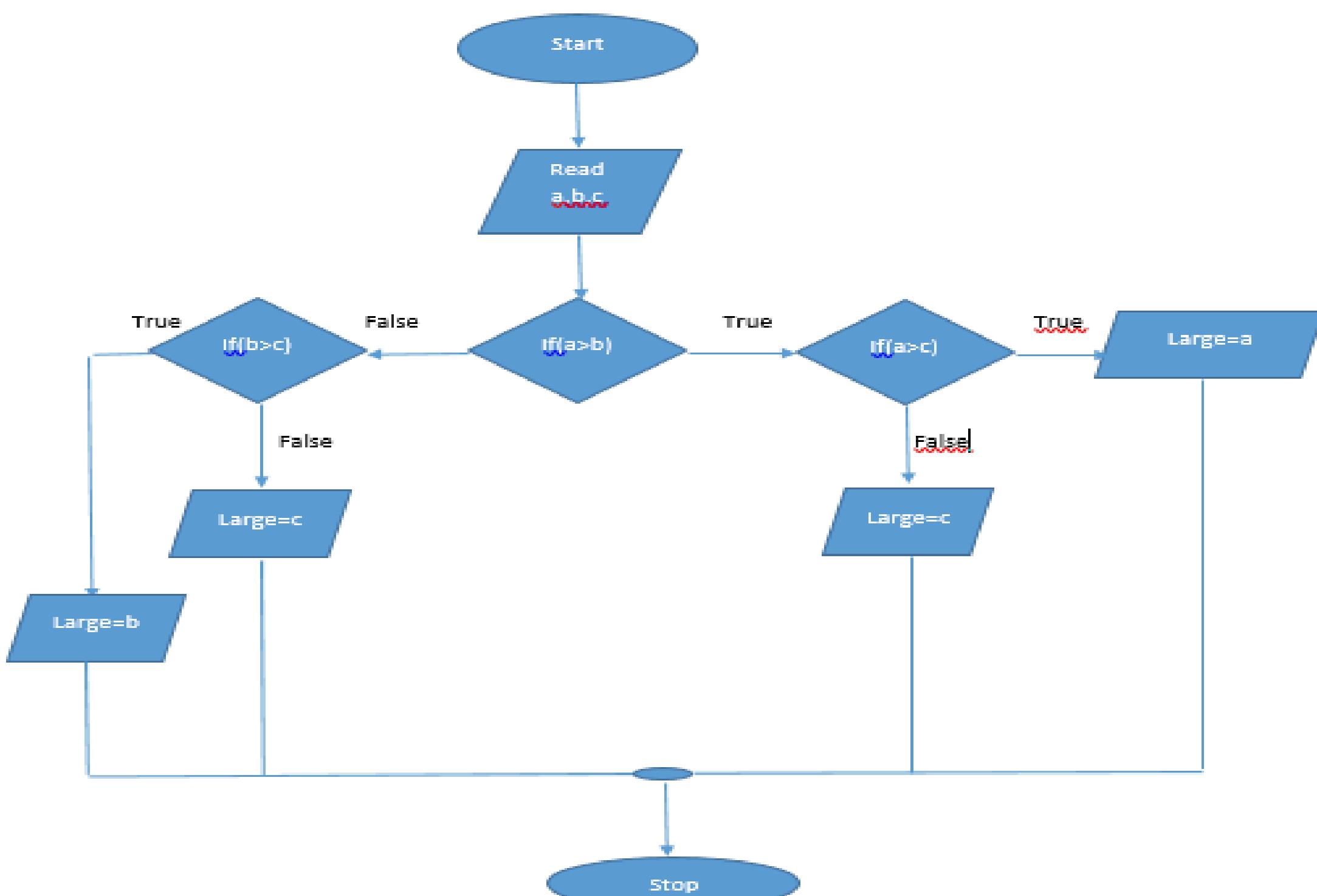
    if( $b > c$ ) Then

large=b.

else

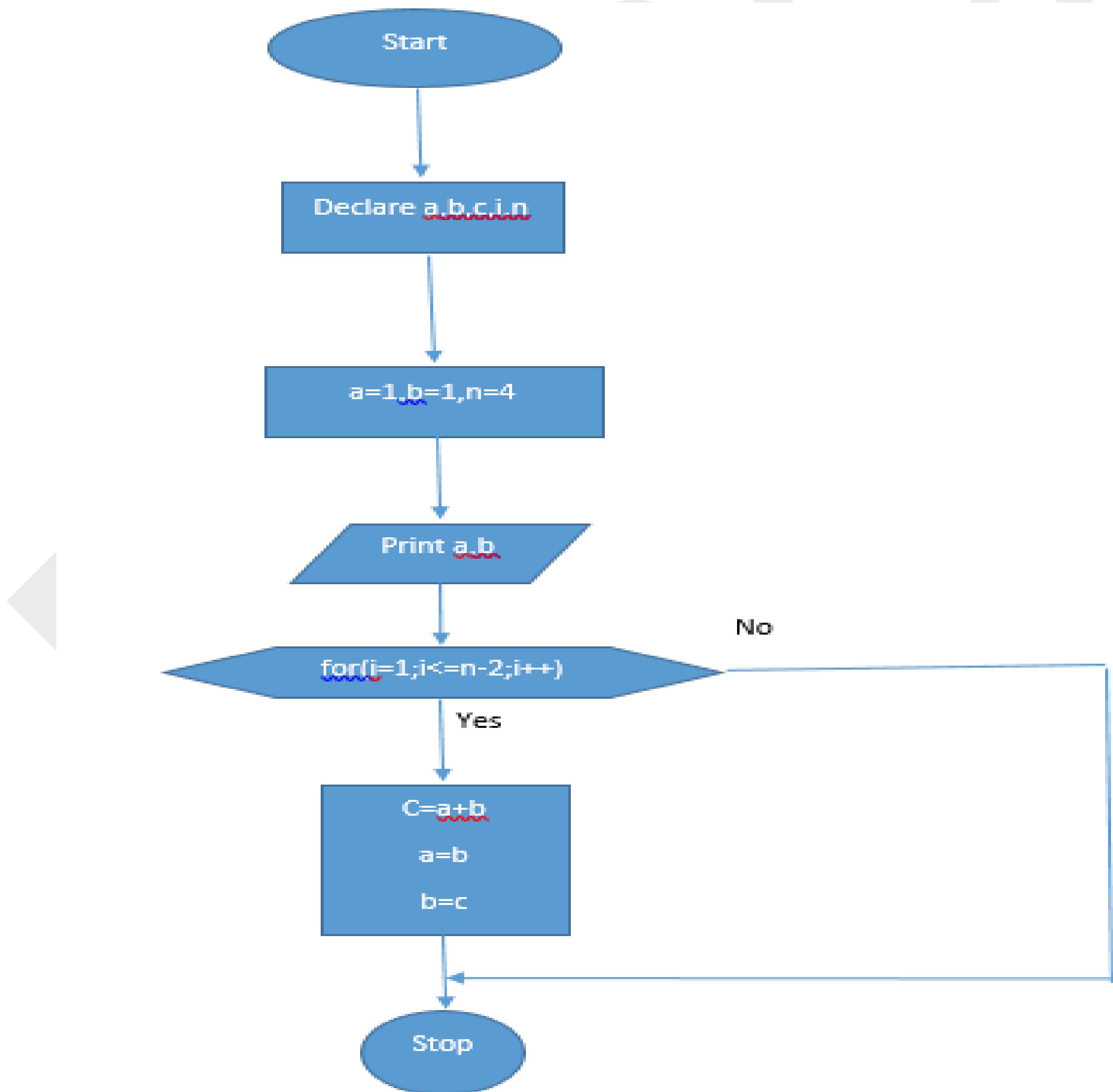
large=c.

end if.

**Step 4:** Display large.**Step 5:** stop.**Flowchart for largest of three numbers.**

**Algorithm for Fibonacci series.**

- Step 1:** Start.  
**Step 2:** Declare variable a,b,c,i,n.  
**Step 3:** Initialize a=1,b=1,n=4.  
**Step 4:** Display a,b.  
**Step 5:** Repeat this until  $i \leq n-2$ .  
**Step 6:**  $c = a + b$ .  
**Step 7:**  $a = b$ .  
**Step 8:**  $b = c$ .  
**Step 9:** Stop.

**Flowchart for Fibonacci series.**

**1. Write a program to display “Hello, World!”**

```
#include<stdio.h>/Preprocessor directives
#include<conio.h>

void main()
{
    clrscr();
    printf("Hello, World! \n"); //Executable statement
    getch();
}
```

**Output****Hello,World!**

**2. Write a program to add two numbers and display the result.**

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

void main()
{
    int a,b,c;
    clrscr();
    printf("enter the values of a and b\n");
    scanf("%d%d",&a,&b);
    c=a+b;
    printf("Addition of two numbers is:%d",c); //Display Result
    getch();
}
```

**Output**

enter the values of a and b  
2,3  
Addition of two numbers is: 5

3. Write a program to demonstrate the use of keywords, identifiers, constants, and variables.

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

// Main function: entry point of the program

int main()
{
    // Keywords: int, float, return
    // Identifiers: num1, num2, sum, average
    // Constants: 100, 3.0
    int a, b;      // Variable declaration
    const float divisor= 3.0; // Constant declaration
    float sum, average; // More variable declarations
    clrscr();

    // Prompt user for input
    printf("Enter two integers: ");
    scanf("%d %d",&a,&b); // Read two integers from user

    // Calculate sum and average
    sum = a+b; // Variable assignment
    average = sum / divisor; // Calculate average

    // Display results
    printf("Sum: %d\n", (int) sum); // Display the sum
    printf("Average: %.2f\n", average);

    // Return statement
    return 0; // Exit status of the program
}
```

**Output**

**Enter two integers: 5 10**

**Sum: 15**

**Average: 5.00**

PBCHBA

4. Write a program to declare and initialize variables of different data types (int, float, char, double).

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

```
#include<conio.h>
```

```
int main()
```

```
{
```

```
    // Declare and initialize variables of different data types
```

```
    int a= 42;          // Integer variable
```

```
    float b= 3.14f;     // Float variable
```

```
    char c = 'A';       // Character variable
```

```
    double d= 2.71828;  // Double variable
```

```
    clrscr();
```

```
    // Print the values of the variables
```

```
    printf("Integer variable: %d\n", a);
```

```
    printf("Float variable: %.2f\n", b);
```

```
    printf("Character variable: %c\n", c);
```

```
    printf("Double variable: %.5f\n", d);
```

```
    return 0;
```

```
}
```

#### Output

Integer variable: 42

Float variable: 3.14

Character variable: A

Double variable: 2.71828

**5. Write a program to define and use symbolic constants.**

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14159 // Using #define to create symbolic constants
#define MAX_SIZE 100

// Using const to create a symbolic constant
const int MIN_SIZE = 1;

int main()
{
    // Using the defined constants
    double radius = 5.0;
    double area;
    clrscr();

    // Calculate the area of a circle using the PI constant
    area = PI * radius * radius;

    printf("Radius: %.2f\n", radius);
    printf("Area of the circle: %.2f\n", area);

    // Print max and min sizes
    printf("Max size: %d\n", MAX_SIZE);
    printf("Min size: %d\n", MIN_SIZE);

    return 0;
}
```

**Output**

Radius: 5.00  
Area of the circle: 78.54  
Max size: 100  
Min size: 1

6. Write a program to demonstrate the use of different control string and escape sequences in printf and scanf.

```
#include<stdio.h>
int main()
{
    // Variables for user input
    char name[50];
    int age;
    float height;
    clrscr();

    // Using escape sequences in printf
    printf("Welcome to the C Programming Demo!\n");
    printf("=====\\n");

    // Using printf control strings
    printf("Please enter your name: ");
    scanf("%s", name); // Limit input to avoid buffer overflow

    printf("Enter your age: ");
    scanf("%d",&age);

    printf("Enter your height in meters (e.g., 1.75): ");
    scanf("%f",&height);

    // Using different control strings in printf
    printf("\\nHello, %s!\\n", name);
    printf("You are %d years old.\\n", age);
    printf("Your height is %.2f meters.\\n", height);

    // Demonstrating more escape sequences
    printf("\\nSummary:\\n");
    printf("=====\\n");
    printf("Name: %s\\n", name);
```

```
printf("Age: %d\n", age);
printf("Height: %.2f meters\n", height);
printf("this is \t tab space\n");
printf("this is alert \a sound");
printf("Thank you for participating!\n");
printf("Goodbye!\n");
getch();
return 0;
}
```

### Output

**Welcome to the C Programming Demo!**

=====

Please enter your name: Raju

Enter your age: 32

Enter your height in meters (e.g., 1.75): 5.4

Hello, Raju!

You are 32 years old.

Your height is 5.40 meters.

**Summary:**

=====

Name: Raju

Age: 32

Height: 5.40 meters

this is tab space

this is alert sound

Thank you for participating!

Goodbye!

7. Write a program to read and display a single character using getchar and putchar.

```
#include<stdio.h>

int main()
{
    char ch;
    clrscr();

    // Prompt the user for input
    printf("Please enter a single character: ");

    // Read a character using getchar
    ch = getchar();

    // Display the character using putchar
    printf("You entered: ");
    putchar(ch);
    printf("\n"); // Newline for better formatting

    return 0;
}
```

**Output**

Please enter a single character: A  
You entered: A

**8. Write a program to read and display a string using gets and puts.**

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

int main()
{
    char str[100]; // Array to hold the string
    clrscr();

    // Prompt the user for input
    printf("Please enter a string: ");

    // Read a string using gets
    gets(str)
    // Display the string using puts
    printf("You entered:");
    puts(str);

    return 0;
}
```

**Output**

Please enter a string: Ballari  
You entered: Ballari

9. Write a program to perform basic arithmetic operations (+,-,\*,/,% ) and assignment, increment, and decrement operators.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c,d,e,f,g;
    clrscr();
    printf("enter the values of a and b\n");
    scanf("%d%d",&a,&b);
    c=a+b;
    d=a-b;
    e=a*b;
    f=a/b;
    g=a%b;
    printf("The sum of two numbers is:%d\n",c);
    printf("The subtraction of two numbers is:%d\n",d);
    printf("The multiplication of two numbers is:%d\n",e);
    printf("The division of two numbers is:%d\n",f);
    printf("The modulus of two numbers is:%d\n",g);

    // Increment and decrement operator.
    a++;
    printf(" After increment the value of a is :%d\n",a);
    b--;
    printf(" After decrement the value of b is :%d\n",b);
    getch();

}
```

**Output**

**enter the values of a and b**

**6 4**

**The sum of two numbers is: 10**

**The subtraction of two numbers is: 2**

**The multiplication of two numbers is: 24**

**The division of two numbers is: 1**

**The modulus of two numbers is: 2**

**After increment the value of a is: 7**

**After decrement the value of b is : 3**

10. Write a program to demonstrate the use of relational and logical operators and bitwise, conditional operators.

```
#include<stdio.h>
int main()
{
    int num1, num2;
    clrscr();

    // Input from the user
    printf("Enter two integers:\n");
    printf("First integer: ");
    scanf("%d",&num1);
    printf("Second integer: ");
    scanf("%d",&num2);

    // Relational Operators
    printf("\nRelational Operator Results:\n");
    printf("%d>%d: %s\n", num1, num2, (num1>num2) ? "True" : "False");
    printf("%d<%d: %s\n", num1, num2, (num1<num2) ? "True" : "False");
    printf("%d == %d: %s\n", num1, num2, (num1 == num2) ? "True" :
"False");
    printf("%d != %d: %s\n", num1, num2, (num1 != num2) ? "True" :
"False");
    printf("%d>= %d: %s\n", num1, num2, (num1>= num2) ? "True" :
"False");
    printf("%d<= %d: %s\n", num1, num2, (num1<= num2) ? "True" :
"False");

    // Logical Operators
    printf("\nLogical Operator Results:\n");
    printf("num1>0 AND num2>0: %s\n", (num1>0&&num2>0) ? "True" :
"False");
    printf("num1>0 OR num2>0: %s\n", (num1>0 || num2>0) ? "True" :
"False");
    printf("NOT (num1>0): %s\n", !(num1>0) ? "True" : "False");
```

```
// Bitwise Operators
printf("\nBitwise Operator Results:\n");
printf("num1&num2 = %d\n", num1&num2);
printf("num1 | num2 = %d\n", num1 | num2);
printf("num1 ^ num2 = %d\n", num1 ^ num2);
printf("~num1 = %d\n", ~num1);
printf("num1<<1 = %d\n", num1<<1);
printf("num1>>1 = %d\n", num1>>1);

// Conditional Operator
int max = (num1 > num2) ? num1 : num2;
printf("\nConditional Operator Result:\n");
printf("The maximum of %d and %d is: %d\n", num1, num2, max);

return 0;
}
```

**Output****Enter two integers:****First integer: 7 6****Second integer:****Relational Operator Results:****7>6: True****7<6: False****7 == 6: False****7 != 6: True****7>= 6: True****7<= 6: False****Logical Operator Results:****num1>0 AND num2>0: True****num1>0 OR num2>0: True****NOT (num1>0): False****Bitwise Operator Results:****num1&num2 = 6****num1 | num2 = 7****num1 ^ num2 = 1****~num1 = -8****num1<<1 = 14****num1>>1 = 3****Conditional Operator Result:****The maximum of 7 and 6 is: 7**

**11. Write a program to find the largest of three numbers using if-else statements.**

```
#include<stdio.h>
int main()
{
    int num1, num2, num3;
    clrscr();

    // Input from the user
    printf("Enter three numbers:\n");
    scanf("%d%d%d",&num1,&num2,&num3);

    // Determine the largest number using if-else statements

    if (num1 >= num2 && num1 >= num3)
    {
        printf("The largest number is: %d\n", num1);
    }
    else if (num2>= num1&&num2>= num3)
    {
        printf("The largest number is: %d\n", num2);
    }
    else
    {
        printf("The largest number is: %d\n", num3);
    }

    return 0;
}
```

**Output**

**Enter three numbers:**

**4 5 6**

**The largest number is:**

12. Write a program to classify a student's grade based on marks using nested if-else and else-if ladder.

```
#include<stdio.h>

int main()
{
    int marks;

    // Input from the user
    printf("Enter the student's marks (0 to 100): ");
    scanf("%d",&marks);

    // Grade classification using nested if-else and else-if ladder
    if (marks < 0 || marks > 100)
        printf("Invalid marks! Please enter marks between 0 and 100.\n");

    else if (marks >= 70)
        printf("Distinction\n");

    else if (marks >=60 && marks<70)

        printf("First Class\n");

    else if (marks >= 40 && marks<60)

        printf("Second Class\n");

    else

        printf("Fail\n");

    return 0;
}
```

**Output**

Enter the student's marks (0 to 100): 80Distinction

**13. Write a program to print the day of the week using switch-case statements.**

```
#include<stdio.h>
int main()
{
    int day;

    // Ask the user to input a number between 1 and 7
    printf("Enter a number between 1 and 7: ");
    scanf("%d",&day);

    // Use switch-case to determine the day of the week
    switch (day)
    {
        case 1:
            printf("Monday\n");
            break;
        case 2:
            printf("Tuesday\n");
            break;
        case 3:
            printf("Wednesday\n");
            break;
        case 4:
            printf("Thursday\n");
            break;
        case 5:
            printf("Friday\n");
            break;
        case 6:
            printf("Saturday\n");
            break;
        case 7:
            printf("Sunday\n");
            break;
        default:
            printf("Invalid input! Please enter a number between 1 and 7.\n");
    }
}
```

```
    }  
    getch();  
    return 0;  
}
```

**Output**

**Enter a number between 1 and 7: 4**

**Thursday**

14. Write a program to print the first 10 natural numbers using a while loop.

```
#include<stdio.h>

int main()
{
    int i = 1; // Initialize the counter variable

    // Use a while loop to print the first 10 natural numbers
    while (i <= 10)
    {
        printf("%d \n", i);
        i++; // Increment the counter
    }

    printf("\n"); // Print a newline at the end
    getch();
    return 0;
}
```

**Output**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

15. Write a program to print the factorial of a number using a do-while loop.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i=1,f=1;
    clrscr();

    printf("\n Enter The Number:");
    scanf("%d",&n);

    //LOOP TO CALCULATE THE FACTORIAL OF A NUMBER
    do
    {
        f=f*i;
        i++;
    }while(i<=n);

    printf("\n The Factorial of %d is %d",n,f);
    getch();
}
```

### Output

Enter The Number:5

The Factorial of 5 is 120

**16. Write a program to generate the Fibonacci series using a for loop.**

```
#include<stdio.h>
int main()
{
    int i, n;

    // initialize first and second terms
    int t1 = 0, t2 = 1;

    // initialize the next term (3rd term)
    int nextTerm = t1 + t2;

    // get no. of terms from user
    printf("Enter the number of terms: ");
    scanf("%d",&n);

    // print the first two terms t1 and t2
    printf("Fibonacci Series: %d, %d, ", t1, t2);

    // print 3rd to nth terms
    for (i = 3; i <= n; ++i) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }
    getch();
    return 0;
}
```

**Output**

Enter the number of terms: 5  
Fibonacci Series: 0, 1, 1, 2, 3,

**17. Write a program to print a multiplication table using nested loops.**

```
#include<stdio.h>
int main()
{
    int i, j, n;

    // Ask the user for the size of the multiplication table
    printf("Enter the size of the multiplication table: ");
    scanf("%d",&n);

    // Outer loop for rows (1 to n)
    for (i = 1; i <= 10; i++)
    {
        // Inner loop for columns (1 to n)
        for (j = 1; j <= n; j++)
        {
            // Print the product
            printf("%d x %d = %d, ", j, i, i * j);
        }
        // Move to the next line after each row
        printf("\n");
    }
    getch();
    return 0;
}
```

**Output**

Enter the size of the multiplication table: 3

1x1 = 1, 2x1 = 2, 3x1 = 3,  
1x2 = 2, 2x2 = 4, 3x2 = 6,  
1x3 = 3, 2x3 = 6, 3x3 = 9,  
1x4 = 4, 2x4 = 8, 3x4 = 12,  
1x5 = 5, 2x5 = 10, 3x5 = 15,  
1x6 = 6, 2x6 = 12, 3x6 = 18,  
1x7 = 7, 2x7 = 14, 3x7 = 21,  
1x8 = 8, 2x8 = 16, 3x8 = 24,  
1x9 = 9, 2x9 = 18, 3x9 = 27, 1x10 = 10, 2x10 = 20, 3x10 = 30,

18. Write a program to read and display elements of a one-dimensional array.

```
#include<stdio.h>

int main()
{
    int n,i;
    intarr[5];
    // Ask the user to enter the number of elements
    printf("Enter the number of elements in the array: ");
    scanf("%d",&n);

    int arr[n]; // Declare an array of size 'n'

    // Input elements of the array
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++)
    {
        printf("Element %d: ", i + 1);
        scanf("%d",&arr[i]);
    }

    // Display the elements of the array
    printf("The elements of the array are:\n");
    for (i = 0; i < n; i++) {
        printf("Element %d: %d\n", i + 1, arr[i]);
    }
    getch();
    return 0;
}
```

**Output**

**Enter the number of elements in the array: 5**

**Enter 5 elements:**

**Element 1: 1**

**Element 2: 4**

**Element 3: 3**

**Element 4: 2**

**Element 5: 5**

**The elements of the array are:**

**Element 1: 1**

**Element 2: 4**

**Element 3: 3**

**Element 4: 2**

**Element 5: 5**

19. Write a program to read and display elements of a two-dimensional array.

```
#include<stdio.h>

int main()
{
    int rows, cols,i,j;
    int array[3][3];
// Ask user for the number of rows and columns
    printf("Enter the number of rows: ");
    scanf("%d",&rows);
    printf("Enter the number of columns: ");
    scanf("%d",&cols);

// Declare a 2D array
    int array[rows][cols];

// Read elements of the array
    printf("Enter the elements of the array:\n");
    for ( i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("Element at position (%d, %d): ", i + 1, j + 1);
            scanf("%d",&array[i][j]);
        }
    }

// Display the elements of the array
    printf("\nThe elements of the 2D array are:\n");
    for ( i = 0; i < rows; i++)
    {
        for ( j = 0; j < cols; j++)
        {
            printf("%d ", array[i][j]);
        }
    }
```

```
    }
    printf("\n");
}
getch();
return 0;
}
```

## Output

**Enter the number of rows: 3**

**Enter the number of columns: 3**

**Enter the elements of the array:**

**Element at position (1, 1): 1 2 3**

**Element at position (1, 2): Element at position (1, 3): Element at position  
(2, 1): 3 4 5**

**Element at position (2, 2): Element at position (2, 3): Element at position  
(3, 1): 4 5 6**

**Element at position (3, 2): Element at position (3, 3):**

**The elements of the 2D array are:**

**1 2 3**

**3 4 5**

**4 5 6**

20. Write a program to find the sum and average of elements in an array.

```
#include<stdio.h>

int main()
{
    int n,i,sum;

    // Ask the user for the number of elements in the array
    printf("Enter the number of elements in the array: ");
    scanf("%d",&n);

    // Declare an array to store the elements
    int arr[n];

    // Read elements of the array
    printf("Enter the elements of the array:\n");
    for ( i = 0; i < n; i++)
    {
        printf("Element %d: ", i + 1);
        scanf("%d",&arr[i]);
    }

    // Calculate the sum of the elements
    int sum = 0;
    for (i = 0; i < n; i++)
    {
        sum += arr[i];
    }

    // Calculate the average
    float average = (float)sum / n;

    // Display the sum and average
    printf("\nSum of elements: %d\n", sum);
```

```
printf("Average of elements: %.2f\n", average);

getch();
return 0;
}
```

**Output**

**Enter the number of elements in the array: 5**

**Enter the elements of the array:**

**Element 1: 3**

**Element 2: 45**

**Element 3: 5**

**Element 4: 6**

**Element 5: 5**

**Sum of elements: 23**

**Average of elements: 4.60**

21. Write a program to demonstrate the use of string handling functions (strlen, strcmp, strcpy, strcat).

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char string1[25],string2[25];
    int L;
    clrscr();
    printf("****performing string length****\n");
    printf("Enter the string:\n");
    scanf("%s",string1);
    L=strlen(string1);
    printf("Length of string1 is %d", L);

    printf("****performing string concatenation****\n");
    printf("Enter the two strings\n");
    scanf("%s %s",string1,string2);
    printf("Resultant string is %s:",strcat(string1,string2));

    printf("****performing string copy operation:\n");
    printf("Enter two strings\n");
    scanf("%s %s",string1,string2);
    printf("Resultant string is %s:",strcpy(string1,string2));

    printf("*performing string comparison operation*\n");
    printf("Enter two strings\n");
    scanf("%s %s",string1,string2);
    if(strcmp(string1,string2)==0)
        printf("strings are equal\n");
    else
        printf("strings are not equal\n");

    getch()
```

}

**Output****\*\*\*\*performing string length\*\*\*\*****Enter the string:****Bellary****Length ofstring1 is 7****\*\*\*performing string concatenation\*\*\*****Enter the two string****Hi Hello****Resultant string is HiHello:\*\*\*****performing string copy operation:****Enter two strings****Hospet****Ballari****Resultant string is Ballari****:\*****performing string comparison operation\*****Enter two strings****Hi hello****strings are not equal**

22. Write a program to find the square of a number using a user-defined function.

```
#include<stdio.h>

// Function to calculate the square of a number
int square(int num)
{
    return num * num;
}

int main()
{
    int number, result;

    // Get user input
    printf("Enter a number: ");
    scanf("%d",&number);

    // Call the user-defined function to find the square
    result = square(number);

    // Display the result
    printf("The square of %d is %d\n", number, result);

    return 0;
}
```

**Output**

Enter a number: 4  
The square of 4 is 16

23. Write a program to calculate the sum of two numbers using function with parameters and return type.

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

```
// Function to calculate the sum of two numbers
```

```
int sum(int a, int b)
```

```
{
```

```
    return a + b; // Return the sum of a and b
```

```
}
```

```
int main()
```

```
{
```

```
    int num1, num2, result;
```

```
// Get user input
```

```
printf("Enter the first number: ");
```

```
scanf("%d",&num1);
```

```
printf("Enter the second number: ");
```

```
scanf("%d",&num2);
```

```
// Call the function to calculate the sum
```

```
result = sum(num1, num2);
```

```
// Display the result
```

```
printf("The sum of %d and %d is %d\n", num1, num2, result);
```

```
getch();
```

```
return 0;
```

```
}
```

#### Output

Enter the first number: 5

Enter the second number: 6

The sum of 5 and 6 is 11

24. Write a program to display a message using a function without parameters and return type.

```
#include<stdio.h>

// Function to display a message
void display()
{
    printf("Hello, welcome to the BBC college!\n");
}

int main()
{
    // Call the function to display the message
    display();
    getch();
    return 0;
}
```

**Output**

Hello, welcome to the BBC college!

25. Write a program to declare and initialize pointers and access the value and address of variables using pointers.

```
#include<stdio.h>

void main()
{
    int *p;
    int var=10;
    clrscr();
    p=&var;
    printf("value of variable var is:%d\n",var);
    printf("\n value of variable var is:%d",*p);
    printf("\n address of variable var is:%p",&var);
    printf("\n address of variable var is:%p",p);
    printf("\n address of pointer p is:%p",&p);
    getch();
}
```

**Output**

value of variable var is:10  
value of variable var is:10

address of variable var is:FFF2  
address of variable var is:FFF2  
address of pointer p is:FFF4

26. Write a program to demonstrate the relationship between pointers and arrays.

```
#include<stdio.h>

int main()
int i;
{
    // Declare an array of 5 integers
    int arr[5] = {10, 20, 30, 40, 50};

    // Declare a pointer to an integer
    int *ptr;

    // Make the pointer point to the first element of the array
    ptr = arr;

    // Access array elements using pointer notation
    printf("Array elements using pointer notation:\n");
    for (i = 0; i < 5; i++) {
        printf("Element %d: %d\n", i, *(ptr + i)); // Pointer arithmetic
    }

    // Access array elements using array notation
    printf("\nArray elements using array notation:\n");
    for (i = 0; i < 5; i++)
    {
        printf("Element %d: %d\n", i, arr[i]); // Array indexing
    }

    // Printing address of array elements
    printf("\nAddresses of array elements:\n");
    for (i = 0; i < 5; i++)
    {
        printf("Address of arr[%d]: %p\n", i, (ptr + i)); // Pointer arithmetic
        gives address
    }
    getch();
}
```

```
    return 0;  
}
```

### Output

#### Array elements using pointer notation:

Element 0: 10  
Element 1: 20  
Element 2: 30  
Element 3: 40  
Element 4: 50

#### Array elements using array notation:

Element 0: 10  
Element 1: 20  
Element 2: 30  
Element 3: 40  
Element 4: 50

#### Addresses of array elements:

Address of arr[0]: 0x7ffc56e8c6f0  
Address of arr[1]: 0x7ffc56e8c6f4  
Address of arr[2]: 0x7ffc56e8c6f8  
Address of arr[3]: 0x7ffc56e8c6fc  
Address of arr[4]: 0x7ffc56e8c700

27. Write a program to perform pointer arithmetic (increment, decrement, addition, subtraction)

```
#include<stdio.h>

int main()
int*ptr1,*ptr2;
{
    // Declare and initialize an array of integers
    int arr[] = {10, 20, 30, 40, 50};

    // Declare a pointer and point it to the first element of the array
    int *ptr = arr;

    // Pointer Increment: Move the pointer to the next element
    printf("Pointer Increment:\n");
    printf("Initial value of ptr: %d\n", *ptr); // Points to arr[0]
    ptr++; // Increment pointer to the next element (arr[1])
    printf("After incrementing, ptr points to: %d\n", *ptr); // Points to arr[1]

    // Pointer Decrement: Move the pointer back to the previous element
    printf("\nPointer Decrement:\n");
    ptr--; // Decrement pointer to the previous element (arr[0])
    printf("After decrementing, ptr points to: %d\n", *ptr); // Points back
    to arr[0]

    // Pointer Addition: Add an integer to the pointer
    printf("\nPointer Addition:\n");
    ptr = arr; // Reset pointer to the start of the array
    printf("Initial value of ptr: %d\n", *ptr); // Points to arr[0]
    ptr = ptr + 3; // Add 3 to the pointer, it now points to arr[3]
    printf("After adding 3, ptr points to: %d\n", *ptr); // Points to arr[3]

    // Pointer Subtraction: Subtract an integer from the pointer
    printf("\nPointer Subtraction:\n");
    ptr = ptr - 2; // Subtract 2 from the pointer, it now points to arr[1]
    printf("After subtracting 2, ptr points to: %d\n", *ptr); // Points to arr[1]
```

```
// Pointer subtraction between two pointers (difference of their  
indices)  
printf("\nPointer Subtraction between two pointers:\n");  
ptr1 = arr; // Points to arr[0]  
ptr2 = arr + 4; // Points to arr[4]  
  
// Subtract the pointers to get the difference (number of elements  
between them)  
printf("Difference between ptr2 and ptr1: %ld\n", ptr2 - ptr1); // Should  
output 4  
  
getch();  
return 0;  
}
```

### Output

#### Pointer Increment:

Initial value of ptr: 10

After incrementing, ptr points to: 20

#### Pointer Decrement:

After decrementing, ptr points to: 10

#### Pointer Addition:

Initial value of ptr: 10

After adding 3, ptr points to: 40

#### Pointer Subtraction:

After subtracting 2, ptr points to: 20

#### Pointer Subtraction between two pointers:

Difference between ptr2 and ptr1: 4

28. Write a C program to define a structure for a student (roll number, name, marks) and demonstrate initialization and access of structure members.

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

// Define the structure for student
struct student
{
    int roll_number;
    char name[50];
    float marks;
};

int main()
{
    // Declare and initialize a structure variable for a student
    struct student s1 = {101, "John Doe", 85.5};

    // Declare another structure variable for a student
    struct student s2;

    // Initialize the second student using scanf (input from user)
    printf("Enter roll number for student 2: ");
    scanf("%d",&s2.roll_number);
    printf("Enter name for student 2: ");
    scanf("%s",s2.name);
    printf("Enter marks for student 2: ");
    scanf("%f",&s2.marks);

    // Access and display the structure members for student 1
    printf("\nStudent 1 Details:\n");
    printf("Roll Number: %d\n", s1.roll_number);
    printf("Name: %s\n", s1.name);
    printf("Marks: %.2f\n", s1.marks);
```

```
// Access and display the structure members for student 2
printf("\nStudent 2 Details:\n");
printf("Roll Number: %d\n", s2.roll_number);
printf("Name: %s\n", s2.name);
printf("Marks: %.2f\n", s2.marks);
getch();
return 0;
}
```

### Output

**Enter roll number for student 2: 3**

**Enter name for student 2: ABC**

**Enter marks for student 2: 56**

#### **Student 1 Details:**

**Roll Number: 101**

**Name: John Doe**

**Marks: 85.50**

#### **Student 2 Details:**

**Roll Number: 3**

**Name: ABC**

**Marks: 56.00**

29. Write a C program to create an array of structures for students and display their details.

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

// Define the structure for student
struct student {
    int roll_number;
    char name[50];
    int marks;
};

int main()
{
    int n,i;

    // Declare an array of structures to hold student details
    struct student students[5];

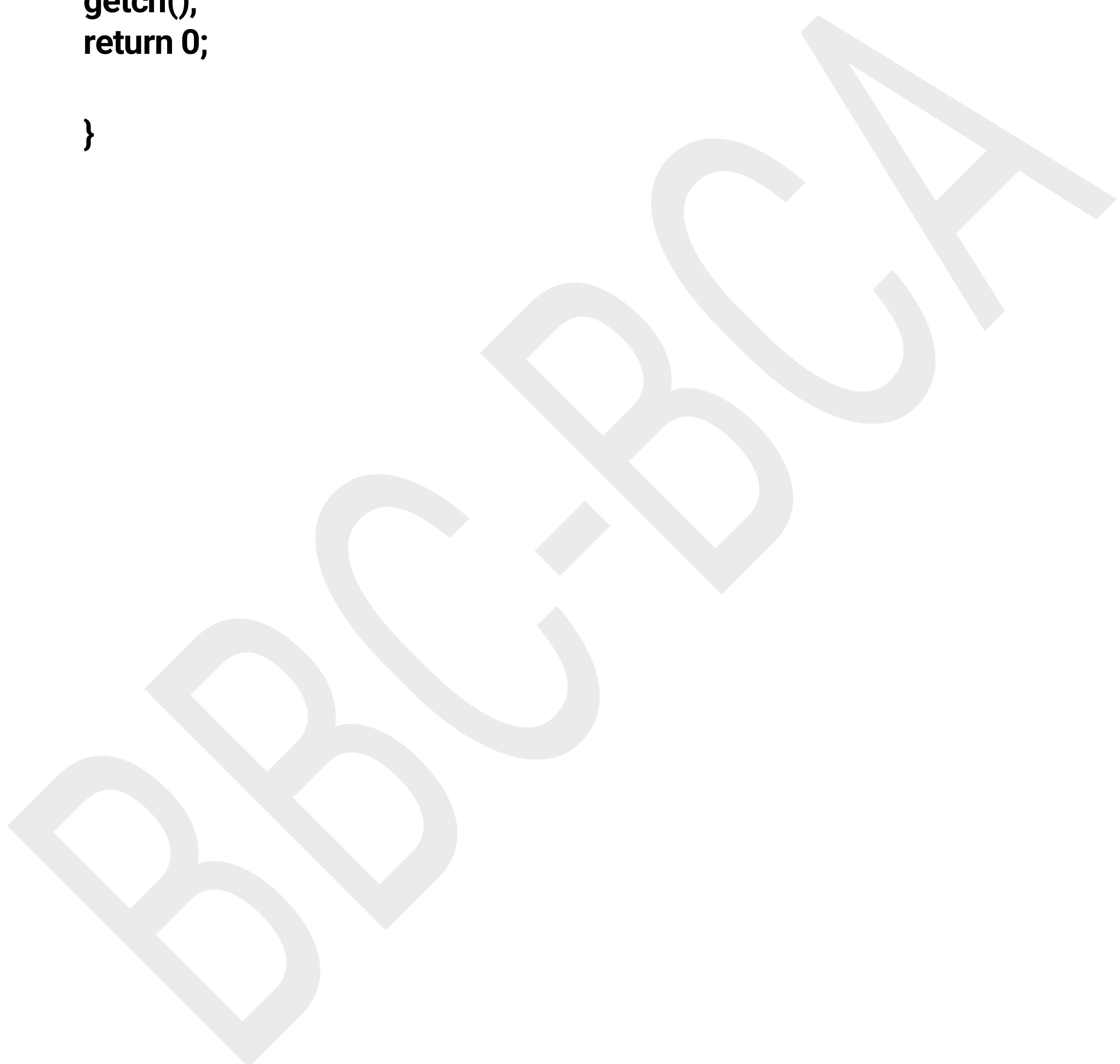
    // Prompt user for the number of students
    printf("Enter the number of students: ");
    scanf("%d",&n);

    // Input details for each student
    for (i = 0; i < n; i++)
    {
        printf("\nEnter details for student %d\n", i + 1);
        printf("Enter roll number: ");
        scanf("%d",&students[i].roll_number);
        printf("Enter name: ");
        scanf("%s", student [i.name]);
        printf("Enter marks: ");
        scanf("%d",&students[i].marks);
    }

    // Display details of each student
    printf("\nStudent Details:\n");
```

```
for (i = 0; i < n; i++)
{
    printf("\nDetails of Student %d:\n", i + 1);
    printf("Roll Number: %d\n", students[i].roll_number);
    printf("Name: %s\n", students[i].name);
    printf("Marks: %d\n", students[i].marks);
}
getch();
return 0;

}
```



**Output**

**Enter the number of students: 2**

**Enter details for student 1**

**Enter roll number: 2**

**Enter name: ABC**

**Enter marks: 66**

**Enter details for student 2**

**Enter roll number: 4**

**Enter name: BBB**

**Enter marks: 77**

**Student Details:**

**Details of Student 1:**

**Roll Number: 2**

**Name: ABC**

**Marks: 66.00**

**Details of Student 2:**

**Roll Number: 4**

**Name: BBB**

**Marks: 77.0**

30. Write a C program to demonstrate embedded structures (structure within a structure).

```
#include<stdio.h>

// Define a structure to store address information
struct Address
{
    char street[50];
    char city[50];
    char state[50];
    int zipCode;
};

// Define a structure to store personal information, which includes an
// Address
struct Person
{
    char name[100];
    int age;
    struct Address address; // Embedded structure
};

int main()
{
    // Declare and initialize a Person structure
    struct Person person1 = { "John Doe", 30,
        {"123 Main St", "New York", "NY", 10001} };
}

// Display the details of person1
printf("Name: %s\n", person1.name);
printf("Age: %d\n", person1.age);
printf("Address: %s, %s, %s - %d\n", person1.address.street,
person1.address.city, person1.address.state,
person1.address.zipCode);
getch();
```

```
    return 0;  
}
```

**Output****Name: Raju****Age: 30****Address: 123 Main St, Ballari, BL – 583104**

PB&BCA

31. Write a C program to define a union for different data types (int, float, char) and demonstrate initialization and access of union members.

```
#include<stdio.h>

// Define a union that can hold an int, float, or char
union Data {
    int i;
    float f;
    char c;
};

int main()
{
    // Declare a union variable
    union Data data;

    // Initialize the union with an integer value
    data.i = 10;
    printf("Data as integer: %d\n", data.i);

    // Initialize the union with a float value
    data.f = 3.14;
    printf("Data as float: %.2f\n", data.f);

    // Initialize the union with a char value
    data.c = 'A';
    printf("Data as char: %c\n", data.c);

    // Access the last value assigned to the union
    printf("Final value stored in the union (char): %c\n", data.c);
    getch();
    return 0;
}
```

**Output****Data as integer: 10****Data as float: 3.14****Data as char: A****Final value stored in the union (char): A**

PB&BCA

32. Write a C program to show the memory usage difference between structures and unions.

```
#include<stdio.h>

// Define a structure that contains an int, float, and char
struct MyStruct
{
    int i;
    float f;
    char c;
};

// Define a union that can hold an int, float, or char
union MyUnion
{
    int i;
    float f;
    char c;
};

int main()
{
    // Declare a variable of type struct MyStruct
    struct MyStruct s;

    // Declare a variable of type union MyUnion
    union MyUnion u;

    // Print the size of the structure and union
    printf("Size of structure: %lu bytes\n", sizeof(s));
    printf("Size of union: %lu bytes\n", sizeof(u));

    getch();
    return 0;
}
```

**Output**

**Size of structure: 12 bytes**

**Size of union: 4 bytes**

PB&BCAF

**33. Write a C Program to Create, Open, Read, Write, and Close a File, and Handle Errors During File Operations**

```
#include<stdio.h>
void main()
{
    FILE *file;
    char filename[] = "example.txt"; // Name of the file
    char writeData[] = "Hello, this is a test file.\n"; // Data to write into the
    file
    char readData[100]; // Buffer to store the data read from the file
    // Part 1: Creating/Opening the file for writing
    file = fopen(filename, "w");
    if (file == NULL)
    {
        perror("Error opening file for writing");
        return 1;
    }
    // Writing data to the file
    if (fputs(writeData, file) == EOF)
    {
        perror("Error writing to file");
        fclose(file);
        return 1;
    }
    fclose(file); // Close the file after writing
    // Part 2: Opening the file for reading
    file = fopen(filename, "r");
    if (file == NULL)
    {
        perror("Error opening file for reading");
        return 1;
    }
    // Reading and displaying the content of the file
    printf("Data read from the file:\n");
    while (fgets(readData, sizeof(readData), file) != NULL)
    {
        printf("%s", readData);
    }
}
```

```
// Checking for read error
if (ferror(file))
{
    perror("Error reading from file");
}
fclose(file);
printf("close the file after reading");
getch();
}
```

**Output**

**Data read from the file:** Hi, this is a test file