A 1 - #include <stdio.h>

```
int add();
int sub();
int div();
int mod();
int multi();


int main() {
    printf("Addition: %d\n", add());
    printf("Subtraction: %d\n",
    sub()); printf("Division: %d\n",
    div());
    printf("Modulo: %d\n", mod());
    printf("Multiplication: %d\n",
    multi()); return 0;
}


int add() {
    int a = 10;
    int b = 5;
    return a + b;
}

int sub() {
    int a = 10;
    int b = 5;
    return a - b;
}

int div() {
    int a = 10;
    int b = 5;
    return a / b;
}

int mod() {
    int a = 10;
    int b = 5;
```

```
    return a % b;
}
int multi() {
    int a = 10;
    int b = 5;
    return a * b;
}
```

A 2 -

```
void swap();
int even();
void marksheet();
float electricity();
int week();
int month();

int main() {

    printf("Let's swap two numbers!\n");
    swap();


    printf("\nIs my number even? %s\n", even() ? "Yes" : "No");


    printf("\nHere is my marksheet:\n");
    marksheet();


    printf("\nElectricity Bill: $%.2f\n", electricity());


    printf("\nWhat day is it today? It's day number %d of the week!\n", week());


    printf("\nWhat's the month? It's month number %d!\n", month());

    return 0;
}


void swap() {
    int a = 10, b = 20, temp;
```

```
    printf("Before Swap: a = %d, b = %d\n", a, b);
    temp = a;
    a = b;
    b = temp;
    printf("After Swap: a = %d, b = %d\n", a, b);
}


int even() {
    int number = 42;
    return (number % 2 == 0);
}


void marksheet() {
    int marks[5] = {85, 90, 78, 88, 76};
    for (int i = 0; i < 5; i++) {
        printf("Subject %d: %d\n", i + 1, marks[i]);
    }
}


float electricity() {
    int units = 250;
    float rate_per_unit = 5.0;
    return units * rate_per_unit;
}


int week() {
    return 3;
}


int month() {
    return 5;
}
```

A 3 - int palindrome();

```
int vowel();
int armstrong();
int reverse_number();
```

```
int sum_of_digits();
int count_digits();
float calculator();
int fibonacci();
int factorial();
int three_number_max();

int main() {
    printf("Palindrome: %s\n", palindrome() ? "Yes" :
    "No"); printf("Vowel: %s\n", vowel() ? "Yes" : "No");
    printf("Armstrong: %s\n", armstrong() ? "Yes" :
    "No"); printf("Reverse Number: %d\n",
    reverse_number()); printf("Sum of Digits: %d\n",
    sum_of_digits());
    printf("Count Digits: %d\n", count_digits());
    printf("Calculator (example of addition): %.2f\n",
    calculator()); printf("Fibonacci: %d\n", fibonacci());
    printf("Factorial: %d\n", factorial());
    printf("Three Number Maximum: %d\n", three_number_max());
    return 0;
}


int palindrome() {
    int num = 121;
    int reversed = 0, original = num, remainder;
    while (num != 0) {
        remainder = num % 10;
        reversed = reversed * 10 + remainder;
        num /= 10;
    }
    return (reversed == original);
}

int vowel() {
    char ch = 'e';
    return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U'); }

int armstrong() {
    int num = 153;
    int sum = 0, temp = num, remainder;
    while (temp != 0) {
```

```
        remainder = temp % 10;
        sum += remainder * remainder * remainder;
        temp /= 10;
    }
    return (sum == num);
}
int reverse_number() {
    int num = 1234;
    int reversed = 0, remainder;
    while (num != 0) {
        remainder = num % 10;
        reversed = reversed * 10 + remainder;
        num /= 10;
    }
    return reversed;
}

int sum_of_digits() {
    int num = 1234;
    int sum = 0, remainder;
    while (num != 0) {
        remainder = num % 10;
        sum += remainder;
        num /= 10;
    }
    return sum;
}

int count_digits() {
    int num = 123456;
    int count = 0;
    while (num != 0) {
        num /= 10;
        count++;
    }
    return count;
}

float calculator() {
    float a = 10.5, b = 5.2;
    char op = '+';
    switch (op) {
```

```
            case '+': return a + b;
            case '-': return a - b;
            case '*': return a * b;
            case '/': return a / b;
            default: return 0;
        }
    }
    int fibonacci() {
        int n = 10;
        int t1 = 0, t2 = 1,
        nextTerm; for (int i = 1; i <
        n; ++i) {
            nextTerm = t1 + t2;
            t1 = t2;
            t2 = nextTerm;
        }
        return t1;
    }

    int factorial() {
        int n = 5;
        int fact = 1;
        for (int i = 1; i <= n; ++i) {
            fact *= i;
        }
        return fact;
    }

    int three_number_max() {
        int a = 10, b = 20, c = 30;
        if (a > b && a > c)
            return a;
        else if (b > a && b > c)
            return b;
        else
            return c;
    }
```

A 4 - #include <stdio.h>

```
    int add(int a, int b);
```

```c
int sub(int a, int b);
float divide(float a, float b);
int mod(int a, int b);
float multi(float a, float b);

int main() {
    int a = 10, b = 5;
    float x = 10.5, y = 5.2;
    printf("Addition: %d\n", add(a, b));
    printf("Subtraction: %d\n", sub(a, b));
    printf("Division: %.2f\n", divide(x, y));
    printf("Modulo: %d\n", mod(a, b));
    printf("Multiplication: %.2f\n", multi(x, y));

    return 0;
}



int add(int a, int b) {
    return a + b;
}

int sub(int a, int b) {
    return a - b;
}

float divide(float a, float b) {
    if (b != 0) {
        return a / b;
    } else {
        printf("Error: Division by zero!\n");
        return 0;
    }
}

int mod(int a, int b) {
    if (b != 0) {
        return a % b;
    } else {
        printf("Error: Division by zero!\n");
        return 0;
    }
```

```
    }

    float multi(float a, float b) {
        return a * b;
    }
```

A 5 -

```
#include <stdio.h>
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    printf("Inside swapp functin: a = %d, b = %d\n", a, b);
}


int even(int number) {
    return number % 2 == 0;
}


float marksheet(int marks[], int size) {
    int total = 0;
    for (int i = 0; i < size; i++) {
        total += marks[i];
    }
    return (float)total / size;
}


float electricity(int units) {
    float cost;
    if (units <= 100) {
        cost = units * 1.5;
    } else if (units <= 200) {
        cost = 100 * 1.5 + (units - 100) * 2.0;
    } else {
        cost = 100 * 1.5 + 100 * 2.0 + (units - 200) * 3.0;
    }
    return cost;
}
```

```
char* week(int day) {
    char* days[] = {"Invalid", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"};
    if (day < 1 || day > 7) {
        return days[0];
    }
    return days[day];
}
char* month(int month) {
    char* months[] = {"Invalid", "January", "February", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December"};
    if (month < 1 || month > 12) {
        return months[0];
    }
    return months[month];
}

int main() {

    int x = 5, y = 10;
    printf("Before swapp: x = %d, y = %d\n", x, y);
    swap(x, y);
    printf("After swapp in main: x = %d, y = %d\n", x, y);


    int num = 4;
    if (even(num)) {
        printf("%d is even\n", num);
    } else {
        printf("%d is odd\n", num);
    }


    int marks[] = {90, 85, 78, 92};
    int size = sizeof(marks) / sizeof(marks[0]);
    float average = marksheet(marks, size);
    printf("Average marks: %.2f\n", average);


    int units = 250;
```

```
    float bill = electricity(units);
    printf("Electicity bill for %d units: $%.2f\n", units, bill);


    int day = 3;
    printf("Day %d is %s\n", day, week(day));


    int month_num = 5;
    printf("Month %d is %s\n", month_num, month(month_num));
    return 0;
}
```

A 6 -

```
#include <stdio.h>



int isPalindrome(int number);
int countVowels(char text[]);
int isArmstrong(int number);
int reverseNumber(int number);
int sumOfDigits(int number);
int countOccurrences(int arr[], int size, int value);
float calculator(float a, float b, char operation);
void fibonacci(int n);
int factorial(int n);
int maxOfThree(int a, int b, int c);

int main() {

    int num = 121;
    if (isPalindrome(num))
        printf("%d is a palindrome\n", num);
    else
        printf("%d is not a palindrome\n", num);


    char text[] = "Hello World";
    printf("Number of vowels: %d\n", countVowels(text));


    num = 153;
    if (isArmstrong(num))
```

```
        printf("%d is an Armstrong number\n",
    num); else
        printf("%d is not an Armstrong number\n", num);


    num = 12345;
    printf("Reversed number: %d\n", reverseNumber(num));


    num = 12345;
    printf("Sum of digits: %d\n", sumOfDigits(num));
    int arr[] = {1, 2, 2, 3, 4, 2, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    int value = 2;
    printf("Number of occurrences of %d: %d\n", value, countOccurrences(arr, size, value));


    float a = 10, b = 5;
    char operation = '+';
    printf("Result: %f\n", calculator(a, b, operation));


    int n = 10;
    printf("Fibonacci sequence: ");
    fibonacci(n);


    n = 5;
    printf("Factorial of %d: %d\n", n, factorial(n));


    int x = 3, y = 7, z = 5;
    printf("Maximum of three numbers: %d\n", maxOfThree(x, y, z));

    return 0;
}


int isPalindrome(int number) {
    int original = number, reversed = 0, remainder;
    while (number != 0) {
        remainder = number % 10;
```

```
        reversed = reversed * 10 + remainder;
        number /= 10;
    }
    return original == reversed;
}

int countVowels(char text[]) {
    int count = 0, i = 0;
    while (text[i]) {
        if (text[i] == 'a' || text[i] == 'e' || text[i] == 'i' || text[i] == 'o' || text[i] == 'u' ||
            text[i] == 'A' || text[i] == 'E' || text[i] == 'I' || text[i] == 'O' || text[i] == 'U')
            {
            count++;
        }
        i++;
    }
    return count;
}

int isArmstrong(int number) {
    int original = number, sum = 0, digits = 0, remainder;
    while (original != 0) {
        digits++;
        original /= 10;
    }
    original = number;
    while (original != 0) {
        remainder = original % 10;
        sum += pow(remainder, digits);
        original /= 10;
    }
    return sum == number;
}

int reverseNumber(int number) {
    int reversed = 0, remainder;
    while (number != 0) {
        remainder = number % 10;
        reversed = reversed * 10 + remainder;
        number /= 10;
    }
    return reversed;
```

```
}

int sumOfDigits(int number) {
    int sum = 0, remainder;
    while (number != 0) {
        remainder = number % 10;
        sum += remainder;
        number /= 10;
    }
    return sum;
}

int countOccurrences(int arr[], int size, int value) {
    int count = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] == value) {
            count++;
        }
    }
    return count;
}

float calculator(float a, float b, char operation) {
    switch (operation) {
        case '+': return a + b;
        case '-': return a - b;
        case '*': return a * b;
        case '/': return a / b;
        default:
            printf("Invalid operation\n");
            exit(1);
    }
}

void fibonacci(int n) {
    int a = 0, b = 1, next;
    for (int i = 0; i < n; i++) {
        if (i <= 1)
            next = i;
        else {
            next = a + b;
            a = b;
```

```
            b = next;
        }
        printf("%d ", next);
    }
    printf("\n");
}

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

int maxOfThree(int a, int b, int c) {
    if (a >= b && a >= c)
        return a;
    else if (b >= a && b >= c)
        return b;
    else
        return c;
}
```

A 7 -

```
#include <stdio.h>


void add();
void sub();
float divide();
int mod(int a, int b);
int multi(int a, int b);

int main() {



    add();


    sub();
```

```
    float divisionResult = divide();
    printf("Division result: %f\n", divisionResult);


    int a = 10, b = 3;
    printf("Modulus of %d and %d: %d\n", a, b, mod(a, b));


    int result = multi(a, b);
    printf("Multiplication of %d and %d: %d\n", a, b, result);

    return 0;
}


void add() {
    int a, b, sum;
    printf("Enter two numbers to add: ");
    scanf("%d %d", &a, &b);
    sum = a + b;
    printf("Sum: %d\n", sum);
}

void sub() {
    int a, b, difference;
    printf("Enter two numbers to subtract: ");
    scanf("%d %d", &a, &b);
    difference = a - b;
    printf("Difference: %d\n", difference);
}

float divide() {
    int a, b;
    printf("Enter two numbers to divide: ");
    scanf("%d %d", &a, &b);
    if (b == 0) {
        printf("Error: Division by
        zero\n"); return 0;
    }
    return (float)a / b;
}
```

```
int mod(int a, int b) {
    return a % b;
}

int multi(int a, int b) {
    return a * b;
}
```