Week – 1

1. #include <stdio.h>

```
int main() {
   float height, base, area;
   printf("Enter the height of the triangle: ");
   scanf("%f", &height);

   printf("Enter the base of the triangle: ");
   scanf("%f", &base);
   area = 0.5 * base * height;
   printf("The area of the triangle is: %.2f\n", area);

   return 0;
}
```

2. #include <stdio.h>

```
int main() {
   float radius, area;
   printf("Enter the radius of the circle: ");
   scanf("%f", &radius);
    area = 3.14159 * radius * radius;
   printf("The area of the circle is: %.2f\n", area);

   return 0;
}
```

3. #include <stdio.h>

```
int main() {
   int marks1, marks2, marks3, lowestMarks;
    printf("Enter marks for student 1: ");
   scanf("%d", &marks1);

   printf("Enter marks for student 2: ");
   scanf("%d", &marks2);

   printf("Enter marks for student 3: ");
   scanf("%d", &marks3);
   lowestMarks = (marks1 < marks2) ? ((marks1 < marks3) ? marks1 : marks3) : ((marks2 < marks3) ? marks2 : marks3);
   printf("The lowest marks among the three students is: %d\n", lowestMarks);

   return 0;
}
```

4. #include <stdio.h>

```
#include <math.h>

int main() {
   float principal, rate, time, compoundInterest;
    printf("Enter the principal amount: ");
   scanf("%f", &principal);
```

```
    printf("Enter the rate of interest (in percentage): ");
    scanf("%f", &rate);

    printf("Enter the time period (in years): ");
    scanf("%f", &time);
     compoundInterest = principal * (pow((1 + rate / 100), time)) - principal;
     printf("The compound interest is: %.2f\n", compoundInterest);

    return 0;
}
```

5.
```
#include <stdio.h>
int calculateCube(int number) {
    return number * number * number;
}

int main() {
    int number, cube;
    printf("Enter a number: ");
    scanf("%d", &number);
    cube = calculateCube(number);
    printf("The cube of %d is: %d\n", number, cube);

    return 0;
}
```

1. #include <stdio.h>

```
int main() {
   int a, b;
    printf("Enter value for a: ");
   scanf("%d", &a);

   printf("Enter value for b: ");
   scanf("%d", &b);
   a = a + b;
   b = a - b;
   a = a - b;
   printf("After interchange, the value of a is: %d\n", a);
   printf("After interchange, the value of b is: %d\n", b);

   return 0;
}
```

2. #include <stdio.h>

```
int main() {
   int a, b;
   printf("Enter value for a: ");
   scanf("%d", &a);

   printf("Enter value for b: ");
   scanf("%d", &b);
   a = a ^ b;
   b = a ^ b;
```

```
    a = a ^ b;
    printf("After interchange, the value of a is: %d\n", a);
    printf("After interchange, the value of b is: %d\n", b);

    return 0;
}
```

3.
```
#include <stdio.h>

int main() {
    printf("Size of int: %lu bytes\n", sizeof(int));
    printf("Size of float: %lu bytes\n", sizeof(float));
    printf("Size of char: %lu bytes\n", sizeof(char));
    printf("Size of double: %lu bytes\n", sizeof(double));
    printf("Size of long double: %lu bytes\n", sizeof(long double));
    printf("Size of short int: %lu bytes\n", sizeof(short int));
    printf("Size of long int: %lu bytes\n", sizeof(long int));
    printf("Size of long long int: %lu bytes\n", sizeof(long long int));

    return 0;
}
```

4.
```
#include <stdio.h>

int isEven(int num) {
    return (num & 1) == 0;
}
```

```
int main() {
    int number;
     printf("Enter a number: ");
    scanf("%d", &number);

    if (isEven(number)) {
        printf("%d is an even number.\n", number);
    } else {
        printf("%d is an odd number.\n", number);
    }

    return 0;
}
```

Week 3

1. #include <stdio.h>

```
int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
     if (number % 2 == 0) {
        printf("%d is an even number.\n", number);
    } else {
        printf("%d is an odd number.\n", number);
    }

    return 0;
```

```
}
```

2. #include <stdio.h>

```
int main() {
   int number;
   printf("Enter a number: ");
   scanf("%d", &number);
   if (number > 0) {
      printf("%d is a positive number.\n", number);
   } else if (number < 0) {
      printf("%d is a negative number.\n", number);
   } else {
      printf("The number is zero.\n");
   }

   return 0;
}
```

3. #include <stdio.h>

```
int main() {
   int year;
   printf("Enter a year: ");
   scanf("%d", &year);
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
      printf("%d is a leap year.\n", year);
   } else {
      printf("%d is not a leap year.\n", year);
```

```
    }

    return 0;
}
```

4. #include <stdio.h>

```
int main() {
    int num1, num2, num3;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
    if (num1 >= num2 && num1 >= num3) {
        printf("%d is the largest number.\n", num1);
    } else if (num2 >= num1 && num2 >= num3) {
        printf("%d is the largest number.\n", num2);
    } else {
        printf("%d is the largest number.\n", num3);
    }

    return 0;
}
```

5. # include <stdio.h>

```
int main() {
    float temperature;
     printf("Enter the temperature in centigrade: ");
    scanf("%f", &temperature);
```

```
    if (temperature < 0) {
        printf("Freezing weather.\n");
    } else if (temperature >= 0 && temperature <= 10) {
        printf("Very Cold weather.\n");
    } else if (temperature > 10 && temperature <= 20) {
        printf("Cold weather.\n");
    } else if (temperature > 20 && temperature <= 30) {
        printf("Normal in Temp.\n");
    } else if (temperature > 30 && temperature <= 40) {
        printf("It's Hot.\n");
    } else {
        printf("It's Very Hot.\n");
    }

    return 0;
}
```

6.

```
#include <stdio.h>

int main() {
    int digit;
    printf("Enter a digit (0 to 9): ");
    scanf("%d", &digit);
    switch (digit) {
        case 0:
            printf("Zero\n");
            break;
        case 1:
            printf("One\n");
```

```
        break;
    case 2:
        printf("Two\n");
        break;
    case 3:
        printf("Three\n");
        break;
    case 4:
        printf("Four\n");
        break;
    case 5:
        printf("Five\n");
        break;
    case 6:
        printf("Six\n");
        break;
    case 7:
        printf("Seven\n");
        break;
    case 8:
        printf("Eight\n");
        break;
    case 9:
        printf("Nine\n");
        break;
    default:
        printf("Invalid input! Please enter a digit between 0 and 9.\n");
}
```

```
    return 0;
}
```

7.
```
#include <stdio.h>

int main() {
    char operator;
    double num1, num2, result;

    printf("Enter an operator (+, -, *, /): ");
    scanf(" %c", &operator);

    printf("Enter two numbers: ");
    scanf("%lf %lf", &num1, &num2);

    switch (operator) {
        case '+':
            result = num1 + num2;
            printf("Result: %.2lf\n", result);
            break;
        case '-':
            result = num1 - num2;
            printf("Result: %.2lf\n", result);
            break;
        case '*':
            result = num1 * num2;
            printf("Result: %.2lf\n", result);
            break;
        case '/':
            if (num2 != 0) {
```

```
            result = num1 / num2;
            printf("Result: %.2lf\n", result);
        } else {
            printf("Error! Division by zero is not allowed.\n");
        }
        break;
    default:
        printf("Invalid operator!\n");
  }

  return 0;
}
```

8.

```
#include <stdio.h>

int main() {
  char choice;
  double area;

  printf("Choose a shape to calculate the area (R for Rectangle, C for Circle, T for Triangle): ");
  scanf(" %c", &choice);

  switch (choice) {
    case 'R':
    case 'r':
      {
        double length, width;
        printf("Enter length and width of the rectangle: ");
```

```
            scanf("%lf %lf", &length, &width);
            area = length * width;
            printf("Area of rectangle: %.2lf\n", area);
        }
        break;
    case 'C':
    case 'c':
        {
            double radius;
            printf("Enter the radius of the circle: ");
            scanf("%lf", &radius);
            area = 3.14159 * radius * radius;
            printf("Area of circle: %.2lf\n", area);
        }
        break;
    case 'T':
    case 't':
        {
            double base, height;
            printf("Enter the base and height of the triangle: ");
            scanf("%lf %lf", &base, &height);
            area = 0.5 * base * height;
            printf("Area of triangle: %.2lf\n", area);
        }
        break;
    default:
        printf("Invalid choice!\n");
}
```

```
    return 0;
}
```

Week 4

1.

```
#include <stdio.h>

int main() {
    int number;

    printf("Enter a number to print its multiplication table: ");
    scanf("%d", &number);

    for (int i = 1; i <= 10; ++i) {
        printf("%d x %d = %d\n", number, i, number * i);
    }

    return 0;
}
```

2.

```
#include <stdio.h>

int main() {
    int number, i;
    unsigned long long factorial = 1;
```

```
    printf("Enter a non-negative integer: ");
    scanf("%d", &number);

    for (i = 1; i <= number; ++i) {
        factorial *= i;
    }

    printf("Factorial of %d = %llu\n", number, factorial);

    return 0;
}
```

3.

```
#include <stdio.h>

int main() {
    int number, reversedNumber = 0, originalNumber, remainder;

    printf("Enter an integer: ");
    scanf("%d", &number);

    originalNumber = number;

    while (number != 0) {
        remainder = number % 10;
        reversedNumber = reversedNumber * 10 + remainder;
        number /= 10;
    }
```

```
    if (originalNumber == reversedNumber)
        printf("%d is a palindrome.\n", originalNumber);
    else
        printf("%d is not a palindrome.\n", originalNumber);

    return 0;
}
```

4.

```
#include <stdio.h>

int main() {
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    printf("Digit frequencies:\n");
    while (number != 0) {
        printf("Digit %d: %d times\n", number % 10, number % 10);
        number /= 10;
    }

    return 0;
}
```

5.

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

```
int main() {
   int num1, num2, i, gcd, lcm;

   printf("Enter two integers: ");
   scanf("%d %d", &num1, &num2);

   for (i = 1; i <= num1 && i <= num2; ++i) {
      if (num1 % i == 0 && num2 % i == 0)
         gcd = i;
   }

   lcm = (num1 * num2) / gcd;

   printf("GCD of %d and %d: %d\n", num1, num2, gcd);
   printf("LCM of %d and %d: %d\n", num1, num2, lcm);

   return 0;
}
```

6.

```
#include <stdio.h>

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

   printf("Enter a number: ");
   scanf("%d", &n);

   printf("Prime numbers between 1 and %d are:\n", n);
```

```
    for (i = 2; i <= n; ++i) {
        int isPrime = 1;

        for (j = 2; j * j <= i; ++j) {
            if (i % j == 0) {
                isPrime = 0;
                break;
            }
        }

        if (isPrime)
            printf("%d\n", i);
    }

    return 0;
}
```

7.

```
#include <stdio.h>

int main() {
    int n, first = 0, second = 1, next;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci series up to %d terms:\n", n);

    for (int i = 1; i <= n; ++i) {
        printf("%d, ", first);
```

```
        next = first + second;
        first = second;
        second = next;
    }

    printf("\n");

    return 0;
}
```

8.

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

int main() {
    int n, originalNumber, remainder, result = 0, digits;

    printf("Enter a number: ");
    scanf("%d", &n);

    originalNumber = n;

    // Count the number of digits
    digits = 0;
    while (originalNumber != 0) {
        originalNumber /= 10;
        ++digits;
    }

    originalNumber = n;
```

```
    // Check if the number is an Armstrong number
    while (originalNumber != 0) {
        remainder = originalNumber % 10;
        result += pow(remainder, digits);
        originalNumber /= 10;
    }

    if (result == n)
        printf("%d is an Armstrong number.\n", n);
    else
        printf("%d is not an Armstrong number.\n", n);
     printf("Armstrong numbers from 1 to %d are:\n", n);
    for (int i = 1; i <= n; ++i) {
        int temp = i, sum = 0;
        while (temp != 0) {
            remainder = temp % 10;
            sum += pow(remainder, digits);
            temp /= 10;
        }
        if (sum == i)
            printf("%d, ", i);
    }

    printf("\n");

    return 0;
}
```

Week 5

1.

```
int main() {
   int N, choice, k, i;
   printf("Enter the size of the array: ");
   scanf("%d", &N);
   int arr[N];
   for (i = 0; i < N; i++) {
      printf("Enter element at position %d: ", i + 1);
      scanf("%d", &arr[i]);
   }
   while (1) {
      printf("\nMenu:\n");
      printf("1. Insert element at kth position\n");
      printf("2. Delete element at kth position\n");
      printf("3. Display array\n");
      printf("4. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
      switch (choice) {
         case 1:
            printf("Enter the position (1 to %d) to insert element: ", N + 1);
            scanf("%d", &k);
            if (k < 1 || k > N + 1) {
               printf("Invalid position. Position should be between 1 and %d.\n", N + 1);
            } else {
               printf("Enter the element to insert: ");
```

```
        int newElement;
        scanf("%d", &newElement);

        for (i = N - 1; i >= k - 1; i--) {
            arr[i + 1] = arr[i];
        }
        arr[k - 1] = newElement;
        N++;
        printf("Element inserted successfully.\n");
    }
    break;
case 2:
    printf("Enter the position (1 to %d) to delete element: ", N);
    scanf("%d", &k);
    if (k < 1 || k > N) {
        printf("Invalid position. Position should be between 1 and %d.\n", N);
    } else {
        for (i = k - 1; i < N - 1; i++) {
            arr[i] = arr[i + 1];
        }
        N--;
        printf("Element deleted successfully.\n");
    }
    break;
case 3:
    printf("Array elements: ");
    for (i = 0; i < N; i++) {
        printf("%d ", arr[i]);
    }
```

```
            printf("\n");
            break;
        case 4:
            printf("Exiting the program.\n");
            return 0;

        default:
            printf("Invalid choice. Please enter a valid option.\n");
      }
   }
   return 0;
}
```

2. Write the program to print the biggest and smallest element in an array.

```
#include <stdio.h>
int main() {
   int N, i;
   printf("Enter the size of the array: ");
   scanf("%d", &N);
   int arr[N];
   for (i = 0; i < N; i++) {
      printf("Enter element at position %d: ", i + 1);
      scanf("%d", &arr[i]);
   }
   int largest = arr[0];
   int smallest = arr[0];
   for (i = 1; i < N; i++) {
      if (arr[i] > largest) {
         largest = arr[i];
```

```
        }
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }
    printf("The largest element in the array is: %d\n", largest);
    printf("The smallest element in the array is: %d\n", smallest);
    return 0;
}
```

3. Write the program to print the sum and average of an array.

```
#include <stdio.h>
int main() {
    int N, i;
    printf("Enter the size of the array: ");
    scanf("%d", &N);
    int arr[N];
    for (i = 0; i < N; i++) {
        printf("Enter element at position %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
    int sum = 0;
    float average;
    for (i = 0; i < N; i++) {
        sum += arr[i];
    }
    average = (float)sum / N;
    printf("The sum of the elements in the array is: %d\n", sum);
    printf("The average of the elements in the array is: %.2f\n", average);
```

```
    return 0;
}
```

4. Write the program to sort an array using bubble sort.

```
#include <stdio.h>
int main() {
    int N, i, j, temp;
    printf("Enter the size of the array: ");
    scanf("%d", &N);
    int arr[N];
    for (i = 0; i < N; i++) {
        printf("Enter element at position %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
    for (i = 0; i < N - 1; i++) {
        for (j = 0; j < N - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("Sorted array: ");
    for (i = 0; i < N; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
```

```
}
```

5. Write the program to search an element using linear search as well as binary search.

```
#include <stdio.h>
int main() {
   int N, i, element;
   printf("Enter the size of the array: ");
   scanf("%d", &N);
   int arr[N];
   printf("Enter the elements of the array:\n");
   for (i = 0; i < N; i++) {
      scanf("%d", &arr[i]);
   }
   printf("\nEnter the element to search using linear search: ");
   scanf("%d", &element);
   int linearIndex = -1;
   for (i = 0; i < N; i++) {
      if (arr[i] == element) {
         linearIndex = i;
         break;
      }
   }
   if (linearIndex != -1) {
      printf("Element %d found at position %d using linear search.\n", element, linearIndex + 1);
   } else {
      printf("Element %d not found in the array using linear search.\n", element);
   }
   printf("\nEnter the element to search using binary search: ");
   scanf("%d", &element);
```

```
    int low = 0, high = N - 1, mid, binaryIndex = -1;
    while (low <= high) {
        mid = (low + high) / 2;
        if (arr[mid] == element) {
            binaryIndex = mid;
            break;
        } else if (arr[mid] < element) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    if (binaryIndex != -1) {
        printf("Element %d found at position %d using binary search.\n", element, binaryIndex + 1);
    } else {
        printf("Element %d not found in the array using binary search.\n", element);
    }
    return 0;
}
```

6. Take an array of 20 integer inputs from user and print the following:

a. number of positive numbers

b. number of negative numbers

c. number of odd numbers

d. number of even numbers e. number of 0.

```
#include <stdio.h>
int main() {
    int arr[20];
```

```
    int positiveCount = 0, negativeCount = 0, oddCount = 0, evenCount = 0, zeroCount = 0;
    printf("Enter 20 integers:\n");
    for (int i = 0; i < 20; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 0; i < 20; i++) {
        if (arr[i] > 0) {
            positiveCount++;
        } else if (arr[i] < 0) {
            negativeCount++;
        } else {
            zeroCount++;
        }
        if (arr[i] % 2 == 0) {
            evenCount++;
        } else {
            oddCount++;
        }
    }
    printf("\na. Number of positive numbers: %d\n", positiveCount);
    printf("b. Number of negative numbers: %d\n", negativeCount);
    printf("c. Number of odd numbers: %d\n", oddCount);
    printf("d. Number of even numbers: %d\n", evenCount);
    printf("e. Number of zeros: %d\n", zeroCount);
    return 0;
}
```

7. Take an array of 10 elements. Split it into middle and store the elements in two different arrays.

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

```
int main(){
    int initialArray[10];
    int firstHalf[5], secondHalf[5];
    printf("Enter 10 integers:\n");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &initialArray[i]);
    }
    for (int i = 0; i < 5; i++) {
        firstHalf[i] = initialArray[i];
        secondHalf[i] = initialArray[i + 5];
    }
    printf("\nINITIAL array: ");
    for (int i = 0; i < 10; i++) {
        printf("%d, ", initialArray[i]);
    }
    printf("\n");
    printf("After splitting:\n");
    printf("First Half: ");
    for (int i = 0; i < 5; i++) {
        printf("%d, ", firstHalf[i]);
    }
    printf("\n");
    printf("Second Half: ");
    for (int i = 0; i < 5; i++) {
        printf("%d, ", secondHalf[i]);
    }
    printf("\n");
    return 0;
}
```

8. Write the program to count frequency of each element in an array.

```
#include <stdio.h>
int main() {
   int N;
   printf("Enter the size of the array: ");
   scanf("%d", &N);
   int arr[N];
   printf("Enter %d integers:\n", N);
   for (int i = 0; i < N; i++) {
      scanf("%d", &arr[i]);
   }
   int frequency[N];
   for (int i = 0; i < N; i++) {
      frequency[i] = 0;
   }
   for (int i = 0; i < N; i++) {
      if (frequency[i] == -1) {
         continue;
      }
      for (int j = i + 1; j < N; j++) {
         if (arr[i] == arr[j]) {
            frequency[j] = -1;
            frequency[i]++;
         }
      }
   }
   printf("\nFrequency of each element:\n");
   for (int i = 0; i < N; i++) {
```

```
        if (frequency[i] != -1) {
            printf("%d occurs %d times.\n", arr[i], frequency[i] + 1);
        }
    }
    return 0;

}
```

Week 6

1.

```
#include <stdio.h>

#define MAX_SIZE 100

void displayArray(int arr[], int size) {
    printf("Array elements: ");
    for (int i = 0; i < size; ++i) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void insertElement(int arr[], int *size, int position, int element) {
    if (*size >= MAX_SIZE) {
        printf("Array is full. Cannot insert more elements.\n");
        return;
    }
```

```
    if (position < 1 || position > *size + 1) {
        printf("Invalid position to insert element.\n");
        return;
    }

    for (int i = *size; i >= position; --i) {
        arr[i] = arr[i - 1];
    }

    arr[position - 1] = element;
    ++*size;

    printf("Element %d inserted at position %d.\n", element, position);
}

void deleteElement(int arr[], int *size, int position) {
    if (*size == 0) {
        printf("Array is empty. Cannot delete element.\n");
        return;
    }

    if (position < 1 || position > *size) {
        printf("Invalid position to delete element.\n");
        return;
    }

    int deletedElement = arr[position - 1];

    for (int i = position - 1; i < *size - 1; ++i) {
```

```
        arr[i] = arr[i + 1];
    }

    --*size;

    printf("Element %d deleted from position %d.\n", deletedElement, position);
}

int main() {
    int arr[MAX_SIZE];
    int size = 0;
    int choice, position, element;

    do {
        printf("Menu:\n");
        printf("1. Insert element\n");
        printf("2. Delete element\n");
        printf("3. Display array\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter position to insert element: ");
                scanf("%d", &position);
                printf("Enter element to insert: ");
                scanf("%d", &element);
                insertElement(arr, &size, position, element);
```

```
            break;
        case 2:
            printf("Enter position to delete element: ");
            scanf("%d", &position);
            deleteElement(arr, &size, position);
            break;
        case 3:
            displayArray(arr, size);
            break;
        case 4:
            printf("Exiting the program.\n");
            break;
        default:
            printf("Invalid choice. Please enter a valid option.\n");
      }

   } while (choice != 4);

   return 0;
}
```

2.

```
#include <stdio.h>

int main() {
   int size;

   printf("Enter the size of the array: ");
   scanf("%d", &size);
```

```
    if (size <= 0) {
        printf("Invalid array size.\n");
        return 1;
    }

    int arr[size];

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

    int smallest = arr[0];
    int largest = arr[0];

    for (int i = 1; i < size; ++i) {
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
        if (arr[i] > largest) {
            largest = arr[i];
        }
    }

    printf("Smallest element: %d\n", smallest);
    printf("Largest element: %d\n", largest);

    return 0;
}
```

3.

```
#include <stdio.h>

int main() {
   int size;

   printf("Enter the size of the array: ");
   scanf("%d", &size);

   if (size <= 0) {
      printf("Invalid array size.\n");
      return 1;
   }

   int arr[size];

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

   int sum = 0;

   for (int i = 0; i < size; ++i) {
      sum += arr[i];
   }

   double average = (double)sum / size;
```

```
    printf("Sum: %d\n", sum);
    printf("Average: %.2lf\n", average);

    return 0;
}
```

4.

```
#include <stdio.h>

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    if (size <= 0) {
        printf("Invalid array size.\n");
        return 1;
    }

    int arr[size];

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

    // Bubble Sort
```

```
    for (int i = 0; i < size - 1; ++i) {
        for (int j = 0; j < size - i - 1; ++j) {
            if (arr[j] > arr[j + 1]) {
                // Swap
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    // Display the sorted array
    printf("Sorted array using Bubble Sort:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

5.

```
#include <stdio.h>

// Linear Search
int linearSearch(int arr[], int size, int key) {
    for (int i = 0; i < size; ++i) {
        if (arr[i] == key) {
            return i; // Element found, return index
        }
```

```
    }
    return -1; // Element not found
}
int binarySearch(int arr[], int size, int key) {
    int low = 0, high = size - 1, mid;

    while (low <= high) {
        mid = (low + high) / 2;

        if (arr[mid] == key) {
            return mid; // Element found, return index
        } else if (arr[mid] < key) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }

    return -1; // Element not found
}

int main() {
    int size, key;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    if (size <= 0) {
        printf("Invalid array size.\n");
```

```
        return 1;
    }

    int arr[size];

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

    printf("Enter the element to search: ");
    scanf("%d", &key);
     int linearIndex = linearSearch(arr, size, key);
    if (linearIndex != -1) {
        printf("Linear Search: Element found at index %d.\n", linearIndex);
    } else {
        printf("Linear Search: Element not found.\n");
    }
     int binaryIndex = binarySearch(arr, size, key);
    if (binaryIndex != -1) {
        printf("Binary Search: Element found at index %d.\n", binaryIndex);
    } else {
        printf("Binary Search: Element not found.\n");
    }

    return 0;
}
```

6.

```c
#include <stdio.h>

int main() {
    int arr[20];
    int positiveCount = 0, negativeCount = 0, oddCount = 0, evenCount = 0, zeroCount = 0;

    printf("Enter 20 integers:\n");

    for (int i = 0; i < 20; ++i) {
        scanf("%d", &arr[i]);

        if (arr[i] > 0) {
            positiveCount++;
        } else if (arr[i] < 0) {
            negativeCount++;
        }

        if (arr[i] % 2 == 0) {
            evenCount++;
        } else {
            oddCount++;
        }

        if (arr[i] == 0) {
            zeroCount++;
        }
    }

    printf("Number of positive numbers: %d\n", positiveCount);
```

```
    printf("Number of negative numbers: %d\n", negativeCount);
    printf("Number of odd numbers: %d\n", oddCount);
    printf("Number of even numbers: %d\n", evenCount);
    printf("Number of 0: %d\n", zeroCount);

    return 0;
}
```

7.

```
#include <stdio.h>

int main() {
    int initialArray[10] = {58, 24, 13, 15, 63, 9, 8, 81, 1, 78};
    int size = 10;

    int middle = size / 2;

    int firstArray[middle];
    int secondArray[size - middle];

    // Splitting the array
    for (int i = 0; i < middle; ++i) {
        firstArray[i] = initialArray[i];
    }

    for (int i = middle; i < size; ++i) {
        secondArray[i - middle] = initialArray[i];
    }

    // Displaying the splitted arrays
```

```
    printf("Initial array:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", initialArray[i]);
    }

    printf("\nAfter splitting:\n");

    printf("First array:\n");
    for (int i = 0; i < middle; ++i) {
        printf("%d ", firstArray[i]);
    }

    printf("\nSecond array:\n");
    for (int i = 0; i < size - middle; ++i) {
        printf("%d ", secondArray[i]);
    }

    printf("\n");

    return 0;
}
```

8.

```
#include <stdio.h>

int main() {
    int initialArray[10] = {58, 24, 13, 15, 63, 9, 8, 81, 1, 78};
    int size = 10;
```

```
int middle = size / 2;

int firstArray[middle];
int secondArray[size - middle];

// Splitting the array
for (int i = 0; i < middle; ++i) {
   firstArray[i] = initialArray[i];
}

for (int i = middle; i < size; ++i) {
   secondArray[i - middle] = initialArray[i];
}

// Displaying the splitted arrays
printf("Initial array:\n");
for (int i = 0; i < size; ++i) {
   printf("%d ", initialArray[i]);
}

printf("\nAfter splitting:\n");

printf("First array:\n");
for (int i = 0; i < middle; ++i) {
   printf("%d ", firstArray[i]);
}

printf("\nSecond array:\n");
for (int i = 0; i < size - middle; ++i) {
```

```
        printf("%d ", secondArray[i]);
    }

    printf("\n");

    return 0;
}
```

Week 7

1.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

int main() {
    int matrix[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    // Row Major Order
    printf("Row Major Order:\n");
    for (int i = 0; i < ROWS; ++i) {
        for (int j = 0; j < COLS; ++j) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
     printf("Column Major Order:\n");
    for (int j = 0; j < COLS; ++j) {
        for (int i = 0; i < ROWS; ++i) {
```

```
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

2.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

int main() {
    int matrix[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int sum = 0;

    for (int i = 0; i < ROWS; ++i) {
        for (int j = 0; j < COLS; ++j) {
            sum += matrix[i][j];
        }
    }

    printf("Sum of the matrix: %d\n", sum);

    return 0;
}
```

3.

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

```
int main() {
    int mat1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int mat2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
    int sum[3][3], product[3][3];

    // Adding two matrices
    for (int i = 0; i < 3; ++i)
        for (int j = 0; j < 3; ++j)
            sum[i][j] = mat1[i][j] + mat2[i][j];

    for (int i = 0; i < 3; ++i)
        for (int j = 0; j < 3; ++j)
            for (int k = 0; k < 3; ++k)
                product[i][j] += mat1[i][k] * mat2[k][j];
     printf("Sum of the matrices:\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j)
            printf("%d ", sum[i][j]);
        printf("\n");
    }

    printf("Product of the matrices:\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j)
            printf("%d ", product[i][j]);
        printf("\n");
    }
```

```
    return 0;
}
```

4.

```
#include <stdio.h>

#define SIZE 3

int main() {
    int matrix[SIZE][SIZE] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int diagonalSum = 0, upperTriangularSum = 0, lowerTriangularSum = 0;

    // Sum of diagonal elements
    for (int i = 0; i < SIZE; ++i) {
        diagonalSum += matrix[i][i];
    }

      for (int i = 0; i < SIZE; ++i) {
        for (int j = i + 1; j < SIZE; ++j) {
            upperTriangularSum += matrix[i][j];
        }
    }

     for (int i = 0; i < SIZE; ++i) {
        for (int j = 0; j < i; ++j) {
            lowerTriangularSum += matrix[i][j];
        }
    }

    printf("Sum of diagonal elements: %d\n", diagonalSum);
```

```
    printf("Sum of upper triangular elements: %d\n", upperTriangularSum);
    printf("Sum of lower triangular elements: %d\n", lowerTriangularSum);

    return 0;
}
```

5.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

int main() {
    int matrix[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int oddFrequency = 0, evenFrequency = 0;

    for (int i = 0; i < ROWS; ++i) {
        for (int j = 0; j < COLS; ++j) {
            if (matrix[i][j] % 2 == 0) {
                evenFrequency++;
            } else {
                oddFrequency++;
            }
        }
    }

    printf("Frequency of odd elements: %d\n", oddFrequency);
    printf("Frequency of even elements: %d\n", evenFrequency);

    return 0;
```

```
}
```

6.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

int main() {
    int matrix[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int rowSum[ROWS] = {0}, colSum[COLS] = {0};

     for (int i = 0; i < ROWS; ++i) {
        for (int j = 0; j < COLS; ++j) {
            rowSum[i] += matrix[i][j];
            colSum[j] += matrix[i][j];
        }
    }


    printf("Sum of each row:\n");
    for (int i = 0; i < ROWS; ++i) {
        printf("Row %d: %d\n", i + 1, rowSum[i]);
    }


    printf("Sum of each column:\n");
    for (int j = 0; j < COLS; ++j) {
        printf("Column %d: %d\n", j + 1, colSum[j]);
    }
```

```
    return 0;
}
```

7.

```
#include <stdio.h>

#define SIZE 3

int main() {
    int matrix[SIZE][SIZE];
    int isDiagonal = 1, isUpperTriangular = 1, isLowerTriangular = 1;

    // Reading the matrix
    printf("Enter the elements of the matrix:\n");
    for (int i = 0; i < SIZE; ++i) {
        for (int j = 0; j < SIZE; ++j) {
            scanf("%d", &matrix[i][j]);
        }
    }


    for (int i = 0; i < SIZE; ++i) {
        for (int j = 0; j < SIZE; ++j) {
            if (i != j && matrix[i][j] != 0) {
                isDiagonal = 0;
            }
            if (i > j && matrix[i][j] != 0) {
                isUpperTriangular = 0;
            }
```

```
            if (i < j && matrix[i][j] != 0) {
                isLowerTriangular = 0;
            }
        }
    }


    if (isDiagonal) {
        printf("The matrix is a diagonal matrix.\n");
    } else if (isUpperTriangular) {
        printf("The matrix is an upper triangular matrix.\n");
    } else if (isLowerTriangular) {
        printf("The matrix is a lower triangular matrix.\n");
    } else {
        printf("The matrix is not a diagonal, upper triangular, or lower triangular matrix.\n");
    }

    return 0;
}
```

8.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

int main() {
    int matrix[ROWS][COLS];
    int isSparse = 1, nonZeroCount = 0;
```

```
    printf("Enter the elements of the matrix:\n");
    for (int i = 0; i < ROWS; ++i) {
        for (int j = 0; j < COLS; ++j) {
            scanf("%d", &matrix[i][j]);
            if (matrix[i][j] != 0) {
                nonZeroCount++;
            }
        }
    }

    // Checking if it's a sparse matrix
    if (nonZeroCount <= (ROWS * COLS) / 2) {
        isSparse = 0;
    }

    if (isSparse) {
        printf("The matrix is a sparse matrix.\n");
    } else {
        printf("The matrix is not a sparse matrix.\n");
    }

    return 0;
}
```

Week 8

1.

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

```
int main() {
    int number = 42;
    int *pointer;



    pointer = &number;

    // Using the pointer
    printf("Value of number: %d\n", number);
    printf("Address of number: %p\n", &number);
    printf("Value of pointer: %d\n", *pointer);
    printf("Address stored in pointer: %p\n", pointer);

    return 0;
}
```

2.

```
#include <stdio.h>

int main() {
    int num1, num2, sum;
    int *ptr1, *ptr2;



    printf("Enter first number: ");
    scanf("%d", &num1);
    printf("Enter second number: ");
    scanf("%d", &num2);
```

```
    ptr1 = &num1;
    ptr2 = &num2;

    sum = *ptr1 + *ptr2;

    // Displaying the result
    printf("Sum of %d and %d is: %d\n", *ptr1, *ptr2, sum);

    return 0;
}
```

3.

```
#include <stdio.h>

int main() {
    int num1, num2;
    int *ptr1, *ptr2, temp;

    // Getting input
    printf("Enter first number: ");
    scanf("%d", &num1);
    printf("Enter second number: ");
    scanf("%d", &num2);

    ptr1 = &num1;
    ptr2 = &num2;

    temp = *ptr1;
    *ptr1 = *ptr2;
```

```
    *ptr2 = temp;



    printf("After swapping, num1 = %d and num2 = %d\n", *ptr1, *ptr2);

    return 0;
}
```

4.

```
#include <stdio.h>

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; ++i) {
        scanf("%d", &(*(arr + i)));
    }

    // Print array elements using pointer
    printf("Array elements are:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", *(arr + i));
    }
```

```
    return 0;
}
```

5.

```
#include <stdio.h>

int main() {
    int size;


    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr1[size], arr2[size];

    printf("Enter the elements of the first array:\n");
    for (int i = 0; i < size; ++i) {
        scanf("%d", &(*(arr1 + i)));
    }


    int *ptr1 = arr1, *ptr2 = arr2;
    for (int i = 0; i < size; ++i) {
        *(ptr2 + i) = *(ptr1 + i);
    }


    printf("Copied array elements are:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", *(arr2 + i));
```

```
    }

    return 0;
}
```

6.

```
#include <stdio.h>

void swapArrays(int *arr1, int *arr2, int size) {
    for (int i = 0; i < size; ++i) {
        // Swap corresponding elements
        int temp = *(arr1 + i);
        *(arr1 + i) = *(arr2 + i);
        *(arr2 + i) = temp;
    }
}

int main() {
    int size;

    printf("Enter the size of the arrays: ");
    scanf("%d", &size);

    int arr1[size], arr2[size];
     printf("Enter the elements of the first array:\n");
    for (int i = 0; i < size; ++i) {
        scanf("%d", &(*(arr1 + i)));
    }
     printf("Enter the elements of the second array:\n");
    for (int i = 0; i < size; ++i) {
```

```
        scanf("%d", &(*(arr2 + i)));
    }


    swapArrays(arr1, arr2, size);

    printf("After swapping:\n");
    printf("First array elements:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", *(arr1 + i));
    }

    printf("\nSecond array elements:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", *(arr2 + i));
    }

    return 0;
}
```

7.

```
#include <stdio.h>

void reverseArray(int *arr, int size) {
    int *start = arr;
    int *end = arr + size - 1;

    while (start < end) {
        // Swap elements pointed by start and end
        int temp = *start;
```

```
        *start = *end;
        *end = temp;
         start++;
        end--;
    }
}

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; ++i) {
        scanf("%d", &(*(arr + i)));
    }



    reverseArray(arr, size);

    printf("Reversed array elements are:\n");
    for (int i = 0; i < size; ++i) {
        printf("%d ", *(arr + i));
    }

    return 0;
}
```

8.

```
#include <stdio.h>

#define ROWS 3
#define COLS 3

void addMatrices(int *mat1, int *mat2, int *result, int rows, int cols) {
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         *(result + i * cols + j) = *(mat1 + i * cols + j) + *(mat2 + i * cols + j);
      }
   }
}

void printMatrix(int *mat, int rows, int cols) {
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         printf("%d ", *(mat + i * cols + j));
      }
      printf("\n");
   }
}

int main() {
   int matrix1[ROWS][COLS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
   int matrix2[ROWS][COLS] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
   int resultMatrix[ROWS][COLS];
   addMatrices(&matrix1[0][0], &matrix2[0][0], &resultMatrix[0][0], ROWS, COLS);
```

```
    printf("Matrix 1:\n");
    printMatrix(&matrix1[0][0], ROWS, COLS);

    printf("\nMatrix 2:\n");
    printMatrix(&matrix2[0][0], ROWS, COLS);

    printf("\nSum of matrices:\n");
    printMatrix(&resultMatrix[0][0], ROWS, COLS);

    return 0;
}
```

9.

```
#include <stdio.h>

#define ROWS1 3
#define COLS1 3
#define ROWS2 3
#define COLS2 3

void multiplyMatrices(int *mat1, int *mat2, int *result, int rows1, int cols1, int rows2, int cols2) {
    for (int i = 0; i < rows1; ++i) {
        for (int j = 0; j < cols2; ++j) {
            *(result + i * cols2 + j) = 0;
            for (int k = 0; k < cols1; ++k) {
                *(result + i * cols2 + j) += *(mat1 + i * cols1 + k) * *(mat2 + k * cols2 + j);
            }
        }
    }
}
```

```
void printMatrix(int *mat, int rows, int cols) {
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         printf("%d ", *(mat + i * cols + j));
      }
      printf("\n");
   }
}

int main() {
   int matrix1[ROWS1][COLS1] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
   int matrix2[ROWS2][COLS2] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
   int resultMatrix[ROWS1][COLS2];
    multiplyMatrices(&matrix1[0][0], &matrix2[0][0], &resultMatrix[0][0], ROWS1, COLS1, ROWS2,
COLS2);
    printf("Matrix 1:\n");
   printMatrix(&matrix1[0][0], ROWS1, COLS1);

   printf("\nMatrix 2:\n");
   printMatrix(&matrix2[0][0], ROWS2, COLS2);

   printf("\nProduct of matrices:\n");
   printMatrix(&resultMatrix[0][0], ROWS1, COLS2);

   return 0;
}
```

Week 9

1.

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

int searchString(char *text, char *pattern) {
    int textLength = strlen(text);
    int patternLength = strlen(pattern);

    for (int i = 0; i <= textLength - patternLength; ++i) {
        int j;
        for (j = 0; j < patternLength; ++j) {
            if (text[i + j] != pattern[j]) {
                break;
            }
        }
        if (j == patternLength) {
            return i; // Pattern found at index i
        }
    }

    return -1; // Pattern not found
}

int main() {
    char text[100], pattern[20];
     printf("Enter the text: ");
    fgets(text, sizeof(text), stdin);
    printf("Enter the pattern: ");
    fgets(pattern, sizeof(pattern), stdin);
    text[strcspn(text, "\n")] = '\0';
```

```
    pattern[strcspn(pattern, "\n")] = '\0';
     int result = searchString(text, pattern);
    if (result != -1) {
        printf("Pattern found at index %d.\n", result);
    } else {
        printf("Pattern not found in the text.\n");
    }

    return 0;
}
2.
#include <stdio.h>
#include <string.h>

void reverseWords(char *str) {
    int start = 0;

    for (int end = 0; str[end] != '\0'; ++end) {
        if (str[end] == ' ') {

            for (int i = end - 1; i >= start; --i) {
                printf("%c", str[i]);
            }
            printf(" ");
            start = end + 1;
        }
    }

    for (int i = strlen(str) - 1; i >= start; --i) {
```

```
        printf("%c", str[i]);
    }
}

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
     str[strcspn(str, "\n")] = '\0';
     reverseWords(str);

    return 0;
}
```

3.

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

int main() {
    char str[100];
    int vowels = 0, consonants = 0, digits = 0, spaces = 0, specialChars = 0;
     printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
     for (int i = 0; str[i] != '\0'; ++i) {
        char ch = tolower(str[i]);

        if (ch >= 'a' && ch <= 'z') {
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
```

```
            vowels++;
        } else {
            consonants++;
        }
    } else if (isdigit(ch)) {
        digits++;
    } else if (isspace(ch)) {
        spaces++;
    } else {
        specialChars++;
    }
  }
   printf("Vowels: %d\n", vowels);
  printf("Consonants: %d\n", consonants);
  printf("Digits: %d\n", digits);
  printf("Spaces: %d\n", spaces);
  printf("Special Characters: %d\n", specialChars);

  return 0;
}
```

4.

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

int main() {
  char str[100];

  printf("Enter a string: ");
  fgets(str, sizeof(str), stdin);
```

```
    str[strcspn(str, "\n")] = '\0';



    printf("Separated characters: ");
    for (int i = 0; str[i] != '\0'; ++i) {
        printf("%c ", str[i]);
    }

    return 0;
}
```

5.

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

int main() {
    char str1[100], str2[100];
     printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    str1[strcspn(str1, "\n")] = '\0';

    printf("Enter the second string: ");
    fgets(str2, sizeof(str2), stdin);
    str2[strcspn(str2, "\n")] = '\0';
     strcat(str1, " ");
    strcat(str1, str2);

    printf("Concatenated string: %s\n", str1);
```

```
    return 0;
}
```

6.

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

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    str[strcspn(str, "\n")] = '\0';
     for (int i = 0; str[i] != '\0'; ++i) {
        if (islower(str[i])) {
            str[i] = toupper(str[i]);
        } else if (isupper(str[i])) {
            str[i] = tolower(str[i]);
        }
    }

    // Display the toggled case string
    printf("Toggled case string: %s\n", str);

    return 0;
}
```

7.

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

```
int areIdentical(char *str1, char *str2) {
    while (*str1 != '\0' && *str2 != '\0') {
        if (*str1 != *str2) {
            return 0;
        }
        str1++;
        str2++;
    }
     if (*str1 == '\0' && *str2 == '\0') {
        return 1; // Identical
    } else {
        return 0; // Not identical
    }
}

int main() {
    char str1[100], str2[100];
    printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    str1[strcspn(str1, "\n")] = '\0';

    printf("Enter the second string: ");
    fgets(str2, sizeof(str2), stdin);
    str2[strcspn(str2, "\n")] = '\0';
     if (areIdentical(str1, str2)) {
        printf("Identical\n");
    } else {
        printf("Not Identical\n");
```

```
    }

    return 0;
}
```

8.

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

int compareStrings(const void *a, const void *b) {
    return strcmp(*(const char **)a, *(const char **)b);
}

int main() {
    int numStudents;

    printf("Enter the number of students: ");
    scanf("%d", &numStudents);
    getchar(); // Consume the newline character

    char **names = (char **)malloc(numStudents * sizeof(char *));
    if (names == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    for (int i = 0; i < numStudents; ++i) {
        printf("Enter the name of student %d: ", i + 1);
        names[i] = (char *)malloc(100 * sizeof(char));
```

```
        fgets(names[i], 100, stdin);
        names[i][strcspn(names[i], "\n")] = '\0'; // Remove newline character
    }

    qsort(names, numStudents, sizeof(char *), compareStrings);
    printf("\nSorted names in alphabetical order:\n");
    for (int i = 0; i < numStudents; ++i) {
        printf("%s\n", names[i]);
        free(names[i]); // Free memory for each name
    }

    free(names); // Free memory for the array of names

    return 0;
}
```

Week 10

1.

```
#include <stdio.h>

int stringLength(char *str) {
    int length = 0;


    while (*str != '\0') {
        length++;
        str++;
    }
```

```
    return length;
}

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    str[strcspn(str, "\n")] = '\0';

    int length = stringLength(str);

    printf("Length of the string: %d\n", length);

    return 0;
}
```

2.

```
#include <stdio.h>

void copyString(char *source, char *destination) {
    while (*source != '\0') {
        *destination = *source;
        source++;
        destination++;
    }
    *destination = '\0';
}
```

```
int main() {
    char source[100], destination[100];

    printf("Enter the source string: ");
    fgets(source, sizeof(source), stdin);

    source[strcspn(source, "\n")] = '\0';



    copyString(source, destination);
    printf("Copied string: %s\n", destination);

    return 0;
}
```

3.

```
#include <stdio.h>

void concatenateStrings(char *str1, char *str2) {
    while (*str1 != '\0') {
        str1++;
    }

    while (*str2 != '\0') {
        *str1 = *str2;
        str1++;
        str2++;
    }

    *str1 = '\0';
```

```
}

int main() {
    char str1[100], str2[100];

    printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    str1[strcspn(str1, "\n")] = '\0';
    printf("Enter the second string: ");
    fgets(str2, sizeof(str2), stdin);
    str2[strcspn(str2, "\n")] = '\0';

    concatenateStrings(str1, str2);

    printf("Concatenated string: %s\n", str1);

    return 0;
}
```

4.

```
#include <stdio.h>

int compareStrings(char *str1, char *str2) {
    while (*str1 != '\0' && *str2 != '\0') {
        if (*str1 != *str2) {
            return 0; // Not equal
        }
        str1++;
        str2++;
    }
```

```
    return (*str1 == '\0' && *str2 == '\0');
}

int main() {
    char str1[100], str2[100];
     printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    str1[strcspn(str1, "\n")] = '\0';

    printf("Enter the second string: ");
    fgets(str2, sizeof(str2), stdin);
    str2[strcspn(str2, "\n")] = '\0';

    if (compareStrings(str1, str2)) {
        printf("Strings are equal.\n");
    } else {
        printf("Strings are not equal.\n");
    }

    return 0;
}
```

5.

```
#include <stdio.h>

void findLargest(int *num1, int *num2, int *num3, int *largest) {
    *largest = (*num1 > *num2) ? ((*num1 > *num3) ? *num1 : *num3) : ((*num2 > *num3) ? *num2 : *num3);
}
```

```
int main() {
   int num1, num2, num3, largest;

   printf("Enter three numbers: ");
   scanf("%d %d %d", &num1, &num2, &num3);

   findLargest(&num1, &num2, &num3, &largest);

   printf("The largest number is: %d\n", largest);

   return 0;
}
```

6.

```
#include <stdio.h>

void findLargest(int *num1, int *num2, int *num3, int *largest) {
   *largest = (*num1 > *num2) ? ((*num1 > *num3) ? *num1 : *num3) : ((*num2 > *num3) ? *num2 : *num3);
}

int main() {
   int num1, num2, num3, largest;

   printf("Enter three numbers: ");
   scanf("%d %d %d", &num1, &num2, &num3);

   findLargest(&num1, &num2, &num3, &largest);
```

```
    printf("The largest number is: %d\n", largest);

    return 0;
}
```

7.

```
#include <stdio.h>

void calculateFactorial(int *num, long long *factorial) {
    *factorial = 1;

    for (int i = 1; i <= *num; ++i) {
        *factorial *= i;
    }
}

int main() {
    int num;
    long long factorial;

    printf("Enter a number: ");
    scanf("%d", &num);

    calculateFactorial(&num, &factorial);

    printf("The factorial of %d is: %lld\n", num, factorial);

    return 0;
}
```

8.

```
#include <stdio.h>

int findLargestEven(int *arr, int size) {
   int largestEven = -1;

   for (int i = 0; i < size; ++i) {
      if (*(arr + i) % 2 == 0 && *(arr + i) > largestEven) {
         largestEven = *(arr + i);
      }
   }

   return largestEven;
}

int main() {
   int size;

   printf("Enter the size of the array: ");
   scanf("%d", &size);
   int arr[size];
   printf("Enter the elements of the array:\n");
   for (int i = 0; i < size; ++i) {
      scanf("%d", &(*(arr + i)));
   }
   int largestEven = findLargestEven(arr, size);
    if (largestEven != -1) {
      printf("The largest even number is: %d\n", largestEven);
   } else {
      printf("No even numbers found in the array.\n");
```

```
    }

    return 0;
}
```

9.

```
#include <stdio.h>

int main() {
    int size;

    // Input array size
    printf("Enter the size of the array: ");
    scanf("%d", &size);

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

    int *ptrArr[size];
    for (int i = 0; i < size; ++i) {
        ptrArr[i] = &arr[i];
    }

    int sum = 0;
    for (int i = 0; i < size; ++i) {
        sum += *ptrArr[i];
    }
```

```
    printf("Sum of elements: %d\n", sum);

    return 0;
}
```

10.

```
#include <stdio.h>

void calculateSimpleInterest(float *principal, float *rate, float *time, float *simpleInterest) {
    *simpleInterest = (*principal * *rate * *time) / 100;
}

int main() {
    float principal, rate, time, simpleInterest;

    printf("Enter the principal amount: ");
    scanf("%f", &principal);

    printf("Enter the rate of interest: ");
    scanf("%f", &rate);

    printf("Enter the time period (in years): ");
    scanf("%f", &time);

    calculateSimpleInterest(&principal, &rate, &time, &simpleInterest);

    printf("Simple Interest: %.2f\n", simpleInterest);

    return 0;
```

```
}
```

11.

```
#include <stdio.h>

int findLargestEven(int *arr, int size) {
    int largestEven = -1;

    for (int i = 0; i < size; ++i) {
        if (*(arr + i) % 2 == 0 && *(arr + i) > largestEven) {
            largestEven = *(arr + i);
        }
    }

    return largestEven;
}

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

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

    int largestEven = findLargestEven(arr, size);
```

```
    if (largestEven != -1) {
        printf("The largest even number is: %d\n", largestEven);
    } else {
        printf("No even numbers found in the array.\n");
    }

    return 0;
}
```

Week 11

1.

```
#include <stdio.h>

int maxOfThree(int a, int b, int c) {
    int max = a;

    if (b > max) {
        max = b;
    }

    if (c > max) {
        max = c;
    }

    return max;
}

int main() {
```

```
    int num1, num2, num3;

    printf("Enter three integers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    int maximum = maxOfThree(num1, num2, num3);

    printf("The maximum of the three integers is: %d\n", maximum);

    return 0;
}
```

2.

```
#include <stdio.h>

int isPrime(int num) {
    if (num <= 1) {
        return 0; // Not prime
    }

    for (int i = 2; i * i <= num; ++i) {
        if (num % i == 0) {
            return 0; // Not prime
        }
    }

    return 1; // Prime
}

int main() {
```

```
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);



    if (isPrime(num)) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }

    return 0;
}
```

3.

```
#include <stdio.h>

unsigned long long factorial(int n) {
    if (n <= 1) return 1;
    return n * factorial(n - 1);
}

int main() {
    int num;

    printf("Enter a non-negative integer: ");
    scanf("%d", &num);

    unsigned long long result = factorial(num);
```

```
    printf("The factorial of %d is: %llu\n", num, result);

    return 0;
}
```

4.

```
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int num1, num2;

    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);

    swap(&num1, &num2);

    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

    return 0;
}
```

5.

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

```
void calculateSumAndAverage(int *arr, int size, int *sum, float *average) {
    *sum = 0;

    for (int i = 0; i < size; ++i) {
        *sum += *(arr + i);
    }

    *average = (float)*sum / size;
}

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);

    int arr[size];

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; ++i) {
        scanf("%d", (arr + i));
    }

    int sum;
    float average;

    calculateSumAndAverage(arr, size, &sum, &average);

    printf("Sum of elements: %d\n", sum);
```

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

    return 0;
}
```

6.

```
#include <stdio.h>

int findGCD(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int num1, num2;

    printf("Enter two non-negative integers: ");
    scanf("%d %d", &num1, &num2);

    int gcd = findGCD(num1, num2);

    printf("GCD of %d and %d is: %d\n", num1, num2, gcd);

    return 0;
}
```

7.

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

int isAlphanumeric(char ch) {
    return isalnum(ch);
}

int isPalindrome(const char *str) {
    int left = 0;
    int right = strlen(str) - 1;

    while (left < right) {
        while (left < right && !isAlphanumeric(str[left])) {
            left++;
        }

        while (left < right && !isAlphanumeric(str[right])) {
            right--;
        }

        if (tolower(str[left]) != tolower(str[right])) {
            return 0;
        }

        left++;
        right--;
    }

    return 1;
```

```
}

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    str[strcspn(str, "\n")] = '\0';

    if (isPalindrome(str)) {
        printf("The string is a valid palindrome.\n");
    } else {
        printf("The string is not a valid palindrome.\n");
    }

    return 0;
}
```

8.

```
#include <stdio.h>

typedef struct {
    float real;
    float imag;
} Complex;

void addComplex(Complex num1, Complex num2, Complex *result) {
    result->real = num1.real + num2.real;
    result->imag = num1.imag + num2.imag;
}
```

```
void subtractComplex(Complex num1, Complex num2, Complex *result) {
    result->real = num1.real - num2.real;
    result->imag = num1.imag - num2.imag;
}

int main() {
    Complex num1, num2, sum, diff;

    printf("Enter the real and imaginary parts of the first complex number: ");
    scanf("%f %f", &num1.real, &num1.imag);

    printf("Enter the real and imaginary parts of the second complex number: ");
    scanf("%f %f", &num2.real, &num2.imag);

    addComplex(num1, num2, &sum);
    subtractComplex(num1, num2, &diff);
     printf("Sum: %.2f + %.2fi\n", sum.real, sum.imag);
    printf("Difference: %.2f + %.2fi\n", diff.real, diff.imag);

    return 0;
}
```

9.

```
#include <stdio.h>

void findSecondLargestAndSmallest(int *arr, int size, int *secondLargest, int *secondSmallest) {
    int largest, smallest;

    if (arr[0] > arr[1]) {
```

```
        largest = arr[0];
        smallest = arr[1];
    } else {
        largest = arr[1];
        smallest = arr[0];
    }

    for (int i = 2; i < size; ++i) {
        if (arr[i] > largest) {
            secondLargest = &largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }

        if (arr[i] < smallest) {
            secondSmallest = &smallest;
            smallest = arr[i];
        } else if (arr[i] < secondSmallest && arr[i] != smallest) {
            secondSmallest = arr[i];
        }
    }
}

int main() {
    int size;

    printf("Enter the size of the array: ");
    scanf("%d", &size);
```

```
    int arr[size];

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

    int secondLargest, secondSmallest;

    findSecondLargestAndSmallest(arr, size, &secondLargest, &secondSmallest);

    printf("Second Largest: %d\n", secondLargest);
    printf("Second Smallest: %d\n", secondSmallest);

    return 0;
}
```

10.

```
#include <stdio.h>

void countOccurrences(int *arr, int size) {
    for (int i = 0; i < size; ++i) {
        if (arr[i] != -1) {
            int count = 1;

            for (int j = i + 1; j < size; ++j) {
                if (arr[j] == arr[i]) {
                    count++;
```

```
                arr[j] = -1;
            }
        }

        printf("Element %d occurs %d times.\n", arr[i], count);
    }
  }
}

int main() {
  int size;

  printf("Enter the size of the array: ");
  scanf("%d", &size);

  int arr[size];

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

  countOccurrences(arr, size);

  return 0;
}
```