Assignment 12

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

void inputArray(int arr[], int size, const char* name);
void printArray(int arr[], int size);
void findMaxMin(int arr[], int size);
void findSum(int arr[], int size);
void printEvenOdd(int arr[], int size);
void printAlternate(int arr[], int size);
int isPrime(int n);
void printPrimes(int arr[], int size);
void sumArrays(int arr1[], int arr2[], int result[], int size);
void mergeArrays(int arr1[], int arr2[], int merged[], int size);
void reverseArray(int arr[], int size);
void sortArray(int arr[], int size);

int main() {
  int sizeMain, sizeSub;

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

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

  // Heap memory allocation
  int* arr = (int*)malloc(sizeMain * sizeof(int));
  int* arr1 = (int*)malloc(sizeSub * sizeof(int));

  int* arr2 = (int*)malloc(sizeSub * sizeof(int));
```

```
int* result = (int*)malloc(sizeSub * sizeof(int));

int* merged = (int*)malloc(2 * sizeSub * sizeof(int));

inputArray(arr, sizeMain, "Main Array");
inputArray(arr1, sizeSub, "Sub Array 1");
inputArray(arr2, sizeSub, "Sub Array 2");

printf("\n----------------------------------------\n");
printArray(arr, sizeMain);

printf("\n----------------------------------------\n");
findMaxMin(arr, sizeMain);

printf("\n----------------------------------------\n");
findSum(arr, sizeMain);

printf("\n----------------------------------------\n");
printEvenOdd(arr, sizeMain);

printf("\n----------------------------------------\n");
printAlternate(arr, sizeMain);

printf("\n----------------------------------------\n");
printPrimes(arr, sizeMain);

printf("\n----------------------------------------\n");
sumArrays(arr1, arr2, result, sizeSub);
printArray(result, sizeSub);

printf("\n----------------------------------------\n");
```

```
    mergeArrays(arr1, arr2, merged, sizeSub);
    printArray(merged, 2 * sizeSub);

    printf("\n----------------------------------------\n");
    reverseArray(arr, sizeMain);
    printArray(arr, sizeMain);

    printf("\n----------------------------------------\n");
    sortArray(arr, sizeMain);
    printArray(arr, sizeMain);

    free(arr);
    free(arr1);
    free(arr2);

    free(result);
    free(merged);

    return 0;
}

void inputArray(int arr[], int size, const char* name) {

    printf("\nEnter elements for %s:\n", name);
    for (int i = 0; i < size; i++) {

        printf("Element [%d]: ", i);

        scanf("%d", &arr[i]);
    }
}
```

```
void printArray(int arr[], int size) {

   for (int i = 0; i < size; i++) {

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

   printf("\n");
}

void findMaxMin(int arr[], int size) {

   int max = arr[0], min = arr[0];
   for (int i = 1; i < size; i++) {

      if (arr[i] > max) max = arr[i];

      if (arr[i] < min) min = arr[i];

   }

   printf("Max: %d\nMin: %d\n", max, min);
}

void findSum(int arr[], int size) {

   int sum = 0;

   for (int i = 0; i < size; i++) {
```

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

    printf("Sum: %d\n", sum);
}

void printEvenOdd(int arr[], int size) {

    printf("Even Elements: ");

    for (int i = 0; i < size; i++) {

        if (arr[i] % 2 == 0)
            printf("%d ", arr[i]);
    }

    printf("\nOdd Elements: ");
    for (int i = 0; i < size; i++) {

        if (arr[i] % 2 != 0)

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

    printf("\n");
}

void printAlternate(int arr[], int size) {

    printf("Elements at Even Indexes: ");
    for (int i = 0; i < size; i += 2){
```

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

    printf("\nElements at Odd Indexes: ");
    for (int i = 1; i < size; i += 2){

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

    printf("\n");
}

int isPrime(int n) {

    if (n < 2) return 0;
    for (int i = 2; i * i <= n; i++){

        if (n % i == 0){

            return 0;
        }
    }

    return 1;
}

void printPrimes(int arr[], int size) {

    printf("Prime Numbers in Array: ");
    for (int i = 0; i < size; i++) {
```

```
        if (isPrime(arr[i])){

            printf("%d ", arr[i]);
        }
    }
    printf("\n");
}

void sumArrays(int arr1[], int arr2[], int result[], int size) {

    for (int i = 0; i < size; i++) {

        result[i] = arr1[i] + arr2[i];
    }
}

void mergeArrays(int arr1[], int arr2[], int merged[], int size) {

    for (int i = 0; i < size; i++) {

        merged[i] = arr1[i];
        merged[i + size] = arr2[i];

    }
}

void reverseArray(int arr[], int size) {

    for (int i = 0; i < size / 2; i++) {
```

```
        int temp = arr[i];
        arr[i] = arr[size - 1 - i];
        arr[size - 1 - i] = temp;

    }
}

void sortArray(int arr[], int size) {

    for (int i = 0; i < size - 1; i++) {

        for (int j = i + 1; j < size; j++) {

            if (arr[i] > arr[j]) {

                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;

            }
        }


    }


}
```