**Exercise-II: Functions , Arrays and String handling**

*Functions:♣*

1. **Write a program to find the factorial of a given number using a function**.
2. #include <stdio.h>
3. long long factorial(int n);
4. int main() {
5. int number;
6. printf("Enter a number: ");
7. scanf("%d", &number);
8. printf("Factorial of %d is: %lld\n", number, factorial(number));
9. return 0;
10. }
11. long long factorial(int n) {
12. if (n == 0 || n == 1) {
13. return 1;
14. } else {
15. return n \* factorial(n - 1);
16. }
17. }

**2. Write a program to find the maximum of two given numbers using a function.**

#include <stdio.h>

int findMax(int num1, int num2);

int main() {

int num1, num2;

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

printf("The maximum number is: %d\n", findMax(num1, num2));

return 0;

}

int findMax(int num1, int num2) {

return (num1 > num2) ? num1 : num2;

}

**3. Write a program to find the sum of all elements in an array using a function.**

#include <stdio.h>

int findSum(int arr[], int size);

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

printf("Sum of array elements: %d\n", findSum(arr, n));

return 0;

}

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

int sum = 0;

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

sum += arr[i];

}

return sum;

}

**4. Write a program to check whether a given number is Armstrong or not using a function.**

#include <stdio.h>

#include <math.h>

int isArmstrong(int number);

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (isArmstrong(num)) {

printf("%d is an Armstrong number.\n", num);

} else {

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

}

return 0;

}

int isArmstrong(int number) {

int originalNumber, remainder, n = 0, result = 0;

originalNumber = number;

while (originalNumber != 0) {

originalNumber /= 10;

++n;

}

originalNumber = number;

while (originalNumber != 0) {

remainder = originalNumber % 10;

result += pow(remainder, n);

originalNumber /= 10;

}

return (result == number);

}

**5. Write a program to find the GCD of two given numbers using a function**.

#include <stdio.h>

int findGCD(int num1, int num2);

int main() {

int num1, num2;

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

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

return 0;

}

int findGCD(int num1, int num2) {

while (num1 != num2) {

if (num1 > num2)

num1 -= num2;

else

num2 -= num1;

}

return num1;

}

**6. Write a program to swap two given numbers using a function**.

#include <stdio.h>

void swapNumbers(int \*num1, int \*num2);

int main() {

int num1, num2;

printf("Enter two numbers: ");

scanf("%d %d", &num1, &num2);

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

swapNumbers(&num1, &num2);

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

return 0;

}

void swapNumbers(int \*num1, int \*num2) {

int temp = \*num1;

\*num1 = \*num2;

\*num2 = temp;

}

**7. Write a program to convert a given decimal number to binary using a function.**

#include <stdio.h>

long decimalToBinary(int decimal);

int main() {

int decimal;

printf("Enter a decimal number: ");

scanf("%d", &decimal);

printf("Binary equivalent: %ld\n", decimalToBinary(decimal));

return 0;

}

long decimalToBinary(int decimal) {

long binary = 0, remainder, base = 1;

while (decimal > 0) {

remainder = decimal % 2;

binary += remainder \* base;

decimal /= 2;

base \*= 10;

}

return binary;

}

**8. Write a program to calculate the area of a circle using a function.**

#include <stdio.h>

#define PI 3.14159265358979323846

double calculateArea(double radius);

int main() {

double radius;

printf("Enter the radius of the circle: ");

scanf("%lf", &radius);

printf("Area of the circle: %lf\n", calculateArea(radius));

return 0;

}

double calculateArea(double radius) {

return PI \* radius \* radius;

}

**9. Write a program to check whether a given number is a palindrome or not using a function**.

#include <stdio.h>

int isPalindrome(int number);

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (isPalindrome(num)) {

printf("%d is a palindrome.\n", num);

} else {

printf("%d is not a palindrome.\n", num);

}

return 0;

}

int isPalindrome(int number) {

int originalNumber, reversedNumber = 0, remainder;

originalNumber = number;

while (originalNumber != 0) {

remainder = originalNumber % 10;

reversedNumber = reversedNumber \* 10 + remainder;

originalNumber /= 10;

}

return (number == reversedNumber);

}

**10. Write a program to calculate the power of a given number using a function.**

#include <stdio.h>

// Function to calculate the power of a number

double calculatePower(double base, int exponent);

int main() {

double base, result;

int exponent;

// Input base and exponent

printf("Enter the base: ");

scanf("%lf", &base);

printf("Enter the exponent: ");

scanf("%d", &exponent);

// Calculate the power using the function

result = calculatePower(base, exponent);

// Display the result

printf("%lf raised to the power of %d is: %lf\n", base, exponent, result);

return 0;

}

// Function to calculate the power of a number

double calculatePower(double base, int exponent) {

double result = 1.0;

// Handle negative exponents

if (exponent < 0) {

base = 1 / base;

exponent = -exponent;

}

// Calculate power using a loop

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

result \*= base;

}

return result;

}

***Arrays:♣***

1. **Write a program to find the largest element in an array.**
2. #include <stdio.h>
3. int findLargestElement(int arr[], int size);
4. int main() {
5. int n;
6. printf("Enter the size of the array: ");
7. scanf("%d", &n);
8. int arr[n];
9. printf("Enter the elements of the array:\n");
10. for (int i = 0; i < n; i++) {
11. scanf("%d", &arr[i]);
12. }
13. printf("Largest element in the array: %d\n", findLargestElement(arr, n));
14. return 0;
15. }
16. int findLargestElement(int arr[], int size) {
17. int max = arr[0];
18. for (int i = 1; i < size; i++) {
19. if (arr[i] > max) {
20. max = arr[i];
21. }
22. }
23. return max;
24. }

**2. Write a program to reverse an array using a loop.**

#include <stdio.h>

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

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

reverseArray(arr, n);

printf("Reversed array: ");

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

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

}

return 0;

}

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

int start = 0, end = size - 1;

while (start < end) {

// Swap elements at start and end indices

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

**3. Write a program to find the second largest element in an array.**

#include <stdio.h>

int findSecondLargest(int arr[], int size);

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

int secondLargest = findSecondLargest(arr, n);

if (secondLargest == -1) {

printf("No second largest element found.\n");

} else {

printf("Second largest element in the array: %d\n", secondLargest);

}

return 0;

}

int findSecondLargest(int arr[], int size) {

if (size < 2) {

return -1; // Array should have at least two elements

}

int largest = arr[0];

int secondLargest = arr[1];

if (largest < secondLargest) {

int temp = largest;

largest = secondLargest;

secondLargest = temp;

}

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];

}

}

return secondLargest;

}

**4. Write a program to find the sum of all even elements in an array.**

#include <stdio.h>

int findSumOfEvenElements(int arr[], int size);

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

int sum = findSumOfEvenElements(arr, n);

printf("Sum of even elements in the array: %d\n", sum);

return 0;

}

int findSumOfEvenElements(int arr[], int size) {

int sum = 0;

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

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

sum += arr[i];

}

}

return sum;

}

**5. Write a program to merge two sorted arrays into a new array.**

#include <stdio.h>

void mergeArrays(int arr1[], int size1, int arr2[], int size2, int result[]);

int main() {

int n1, n2;

printf("Enter the size of the first array: ");

scanf("%d", &n1);

int arr1[n1];

printf("Enter the elements of the first array (sorted):\n");

for (int i = 0; i < n1; i++) {

scanf("%d", &arr1[i]);

}

printf("Enter the size of the second array: ");

scanf("%d", &n2);

int arr2[n2];

printf("Enter the elements of the second array (sorted):\n");

for (int i = 0; i < n2; i++) {

scanf("%d", &arr2[i]);

}

// Assuming the result array is large enough to hold both arrays

int result[n1 + n2];

mergeArrays(arr1, n1, arr2, n2, result);

printf("Merged array: ");

for (int i = 0; i < n1 + n2; i++) {

printf("%d ", result[i]);

}

return 0;

}

void mergeArrays(int arr1[], int size1, int arr2[], int size2, int result[]) {

int i = 0, j = 0, k = 0;

while (i < size1 && j < size2) {

if (arr1[i] < arr2[j]) {

result[k++] = arr1[i++];

} else {

result[k++] = arr2[j++];

}

}

while (i < size1) {

result[k++] = arr1[i++];

}

while (j < size2) {

result[k++] = arr2[j++];

}

}

**6. Write a program to remove duplicates from an array.**

#include <stdio.h>

int removeDuplicates(int arr[], int size);

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

int newSize = removeDuplicates(arr, n);

printf("Array after removing duplicates: ");

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

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

}

return 0;

}

int removeDuplicates(int arr[], int size) {

if (size == 0 || size == 1) {

return size; // No duplicates to remove

}

int j = 0;

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

if (arr[i] != arr[i + 1]) {

arr[j++] = arr[i];

}

}

arr[j++] = arr[size - 1];

return j;

}

7. Write **a program to rotate an array to the left by a given number of positions.**

#include <stdio.h>

void rotateLeft(int arr[], int size, int positions);

int main() {

int n, positions;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

printf("Enter the number of positions to rotate left: ");

scanf("%d", &positions);

rotateLeft(arr, n, positions);

printf("Array after rotating left by %d positions: ", positions);

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

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

}

return 0;

}

void rotateLeft(int arr[], int size, int positions) {

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

int temp = arr[0];

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

arr[j] = arr[j + 1];

}

arr[size - 1] = temp;

}

}

**8. Write a program to find the frequency of each element in an array.**

#include <stdio.h>

void findFrequency(int arr[], int size);

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

findFrequency(arr, n);

return 0;

}

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

int visited[size];

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

visited[i] = -1;

}

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

int count = 1;

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

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

count++;

visited[j] = 0;

}

}

if (visited[i] != 0) {

visited[i] = 1;

printf("Frequency of %d: %d\n", arr[i], count);

}

}

}

9**. Write a program to insert an element into an array at a given position.**

#include <stdio.h>

void insertElement(int arr[], int \*size, int position, int element);

int main() {

int n, position, element;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

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

}

printf("Enter the position to insert the element: ");

scanf("%d", &position);

printf("Enter the element to insert: ");

scanf("%d", &element);

insertElement(arr, &n, position, element);

printf("Array after insertion: ");

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

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

}

return 0;

}

void insertElement(int arr[], int \*size, int position, int element) {

if (position < 0 || position > \*size) {

printf("Invalid position for insertion.\n");

return;

}

(\*size)++;

for (int i = \*size - 1; i > position; i--) {

arr[i] = arr[i - 1];

}

arr[position] = element;

}

***Strings:♣***

1. **Write a program to find the length of a given string using the strlen() function**.
2. #include <stdio.h>
3. #include <string.h>
4. int main() {
5. char str[100];
6. printf("Enter a string: ");
7. gets(str);
8. int length = strlen(str);
9. printf("Length of the string: %d\n", length);
10. return 0;
11. }

**2. Write a program to concatenate two strings using the strcat() function.**

#include <stdio.h>

#include <string.h>

int main() {

char str1[100], str2[100];

printf("Enter the first string: ");

gets(str1);

printf("Enter the second string: ");

gets(str2);

strcat(str1, str2);

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

return 0;

}

3**. Write a program to copy one string to another using the strcpy() function.**

#include <stdio.h>

#include <string.h>

int main() {

char source[100], destination[100];

printf("Enter a string: ");

gets(source);

strcpy(destination, source);

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

return 0;

}

**4. Write a program to compare two strings using the strcmp() function.**

#include <stdio.h>

#include <string.h>

int main() {

char str1[100], str2[100];

printf("Enter the first string: ");

gets(str1);

printf("Enter the second string: ");

gets(str2);

int result = strcmp(str1, str2);

if (result == 0) {

printf("Both strings are equal.\n");

} else if (result < 0) {

printf("First string is lexicographically smaller than the second string.\n");

} else {

printf("First string is lexicographically greater than the second string.\n");

}

return 0;

}

**5. Write a program to convert a given string to uppercase using the toupper() function.**

**#include <stdio.h>**

#include <ctype.h>

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

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

str[i] = toupper(str[i]);

}

printf("Uppercase string: %s\n", str);

return 0;

}

**6. Write a program to convert a given string to lowercase using the tolower() function.**

#include <stdio.h>

#include <ctype.h>

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

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

str[i] = tolower(str[i]);

}

printf("Lowercase string: %s\n", str);

return 0;

}

**7. Write a program to reverse a given string using a loop.**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

int length = strlen(str);

printf("Reversed string: ");

for (int i = length - 1; i >= 0; i--) {

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

}

printf("\n");

return 0;

}

**8. Write a program to find the occurrence of a given character in a string using a loop**.

#include <stdio.h>

int main() {

char str[100];

char target;

printf("Enter a string: ");

gets(str);

printf("Enter the character to find: ");

scanf(" %c", &target);

int count = 0;

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

if (str[i] == target) {

count++;

}

}

printf("Occurrence of '%c' in the string: %d\n", target, count);

return 0;

}

**9. Write a program to split a given string into words using the strtok() function.**

#include <stdio.h>

#include <string.h>

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

printf("Words in the string:\n");

char \*token = strtok(str, " ");

while (token != NULL) {

printf("%s\n", token);

token = strtok(NULL, " ");

}

return 0;

}

**10. Write a program to remove all the spaces from a given string using a loop.**

#include <stdio.h>

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

int i, j;

for (i = 0; str[i] != '\0'; i++) {

if (str[i] != ' ') {

str[j++] = str[i];

}

}

str[j] = '\0';

printf("String after removing spaces: %s\n", str);

return 0;

}