Qno.1)

#include <stdio.h>

int main() {

int row1, col1, row2, col2, i, j, k;

int mat1[10][10], mat2[10][10], result[10][10];

// Get the size of the two matrices from the user

printf("Enter 2-dimensional array size: ");

scanf("%d %d", &row1, &col1);

printf("Enter numbers for each element in two arrays:\n");

// Get the elements of the first matrix from the user

for (i = 0; i < row1; i++) {

for (j = 0; j < col1; j++) {

scanf("%d", &mat1[i][j]);

}

}

// Get the size of the second matrix from the user

printf("Enter 2-dimensional array size: ");

scanf("%d %d", &row2, &col2);

// Get the elements of the second matrix from the user

for (i = 0; i < row2; i++) {

for (j = 0; j < col2; j++) {

scanf("%d", &mat2[i][j]);

}

}

// Check if the matrices can be multiplied

if (col1 != row2) {

printf("Cannot multiply the two matrices. Invalid dimensions.\n");

return 0;

}

// Multiply the matrices and store the result in the result matrix

for (i = 0; i < row1; i++) {

for (j = 0; j < col2; j++) {

result[i][j] = 0;

for (k = 0; k < col1; k++) {

result[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

// Print the result matrix

printf("Result after multiplication of two arrays:\n");

for (i = 0; i < row1; i++) {

for (j = 0; j < col2; j++) {

printf("%4d", result[i][j]);

}

printf("\n");

}

return 0;

}

Qno.2)

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int a, b;

// Get the two numbers from the user

printf("Enter two numbers a & b: ");

scanf("%d %d", &a, &b);

// Swap the numbers using pointers

swap(&a, &b);

// Print the swapped numbers

printf("After swapping by pointer, a = %d, b = %d\n", a, b);

return 0;

}

Qno.3)

#include <stdio.h>

int main() {

char str[100];

printf("Enter any string: ");

gets(str);

printf("Result: ");

puts(str);

return 0;

}

Qno.4)

#include <stdio.h>

int factorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n-1);

}

}

int main() {

int n, fact;

// Get the number from the user

printf("Enter a number: ");

scanf("%d", &n);

// Calculate the factorial using recursion

fact = factorial(n);

// Print the result

printf("Factorial = %d\n", fact);

return 0;

}

Qno.5)

#include <stdio.h>

#include <string.h>

void reverse(char \*str) {

int i, j;

char temp;

j = strlen(str) - 1;

for (i = 0; i < j; i++, j--) {

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

int main() {

char str[100], rev\_str[100];

// Get the string from the user

printf("Enter a string: ");

fgets(str, 100, stdin);

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

// Copy the string to a new string and reverse it

strcpy(rev\_str, str);

reverse(rev\_str);

// Compare the original string and the reversed string

if (strcmp(str, rev\_str) == 0) {

printf("yes, it is a palindrome string\n");

} else {

printf("not a palindrome string\n");

}

return 0;

}

Qno.6)

#include <stdio.h>

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

int i, j, k;

// Copy the elements of arr1 to merged array

for (i = 0; i < size1; i++) {

merged[i] = arr1[i];

}

k = size1;

// Check for repeating elements in arr2 and copy only unique elements to merged array

for (i = 0; i < size2; i++) {

int is\_repeating = 0;

for (j = 0; j < size1; j++) {

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

is\_repeating = 1;

break;

}

}

if (!is\_repeating) {

merged[k] = arr2[i];

k++;

}

}

}

int main() {

int size1, size2, i;

int arr1[100], arr2[100], merged[200];

// Get the size and elements of first array

printf("Enter the size of 1st array: ");

scanf("%d", &size1);

printf("Enter each element: ");

for (i = 0; i < size1; i++) {

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

}

// Get the size and elements of second array

printf("Enter the size of 2nd array: ");

scanf("%d", &size2);

printf("Enter each element: ");

for (i = 0; i < size2; i++) {

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

}

// Merge the arrays and print the result

mergeArrays(arr1, size1, arr2, size2, merged);

printf("Result of merging: ");

for (i = 0; i < size1+size2; i++) {

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

}

printf("\n");

return 0;

}

Qno.7)

#include <stdio.h>

int factorial(int n) {

int i, fact = 1;

for (i = 1; i <= n; i++) {

fact \*= i;

}

return fact;

}

int binomialCoefficient(int n, int k) {

return factorial(n) / (factorial(k) \* factorial(n - k));

}

void printBinomialCoefficients(int order) {

int i, j;

for (i = 0; i <= order; i++) {

for (j = 0; j <= i; j++) {

printf("%d ", binomialCoefficient(i, j));

}

printf("\n");

}

}

int main() {

int order;

printf("Enter the order of binomial power expression: ");

scanf("%d", &order);

printf("Results of printing on monitor:\n");

printBinomialCoefficients(order);

return 0;

}

Qno.8)

#include <stdio.h>

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

int i, j, k;

// Sort the array using bubble sort

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

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

if (arr[j] > arr[j+1]) {

int temp = arr[j];

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

arr[j+1] = temp;

}

}

}

// Remove duplicates by shifting array

k = 0;

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

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

arr[k] = arr[i];

k++;

}

}

// Print the array without duplicates

printf("Results of removing duplicate:");

for (i = 0; i < k; i++) {

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

}

printf("\n");

}

int main() {

int size, i;

int arr[100];

// Get the size and elements of the array from the user

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

scanf("%d", &size);

printf("Enter numbers for each element:");

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

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

}

// Remove duplicates from the array and print the result

removeDuplicates(arr, size);

return 0;

}

Qno.9)

#include <stdio.h>

int kthSmallest(int arr[], int size, int k) {

int i, j, temp;

// Sort the array using selection sort

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

int minIndex = i;

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

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

temp = arr[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

// Return the kth smallest element

return arr[k-1];

}

int main() {

int size, i, k, kthSmallestElem;

int arr[100];

// Get the size and elements of the array from the user

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

scanf("%d", &size);

printf("Enter numbers for each element:");

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

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

}

// Get the desired kth smallest element from the user

printf("Enter desired kth smallest element:");

scanf("%d", &k);

// Find the kth smallest element in the array

kthSmallestElem = kthSmallest(arr, size, k);

// Print the result

printf("Result of %dth smallest element: %d\n", k, kthSmallestElem);

return 0;

}

Qno.10)

#include <stdio.h>

#include <math.h>

double calcMean(int arr[], int size) {

int i, sum = 0;

double mean;

// Calculate the sum of array elements

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

sum += arr[i];

}

// Calculate the mean of array elements

mean = (double) sum / size;

return mean;

}

double calcStdDev(int arr[], int size) {

int i;

double mean, sum = 0, stdDev;

// Calculate the mean of array elements

mean = calcMean(arr, size);

// Calculate the sum of squared deviations from the mean

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

sum += pow(arr[i] - mean, 2);

}

// Calculate the standard deviation

stdDev = sqrt(sum / size);

return stdDev;

}

int main() {

int size, i;

int arr[100];

// Get the size and elements of the array from the user

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

scanf("%d", &size);

printf("Enter numbers for each element:");

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

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

}

// Calculate the standard deviation of the array and print the result

printf("Result of Standard Deviation: %lf\n", calcStdDev(arr, size));

return 0;

}