1. 1 D Char array

Write a C program using functions and pointers for the following

1. Read and display a string

2. Without using string builtin functions, calculate the string length

3. Without using string builtin functions, reverse the string

4. Without using string builtin functions, copy one string into other

5. Read a string and check whether the given character is present or not. If present, count the number of times, it is repeated

#include <stdio.h>

#include <string.h>

// Function to read a string

void readString(char \*str) {

printf("Enter a string: ");

gets(str);

}

// Function to display a string

void displayString(const char \*str) {

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

}

// Function to calculate the string length

int stringLength(const char \*str) {

int length = 0;

while (\*str != '\0') {

length++;

str++;

}

return length;

}

// Function to reverse the string

void reverseString(char \*str) {

int length = stringLength(str);

char \*start = str;

char \*end = str + length - 1;

while (start < end) {

char temp = \*start;

\*start = \*end;

\*end = temp;

start++;

end--;

}

}

// Function to copy one string into another

void copyString(char \*dest, const char \*src) {

while (\*src != '\0') {

\*dest = \*src;

dest++;

src++;

}

\*dest = '\0';

}

// Function to check for the presence of a character and count its occurrences

int countCharacter(const char \*str, char ch) {

int count = 0;

while (\*str != '\0') {

if (\*str == ch) {

count++;

}

str++;

}

return count;

}

int main() {

char inputString[100];

char copyString[100];

char searchChar;

int charCount;

// Read and display a string

readString(inputString);

displayString(inputString);

// Calculate the string length

int length = stringLength(inputString);

printf("String Length: %d\n", length);

// Reverse the string

reverseString(inputString);

printf("Reversed String: %s\n", inputString);

// Copy one string into another

copyString(copyString, inputString);

printf("Copied String: %s\n", copyString);

// Read a string and check for the presence of a character

printf("Enter a character to search: ");

scanf(" %c", &searchChar);

charCount = countCharacter(inputString, searchChar);

if (charCount > 0) {

printf("Character '%c' is present %d times in the string.\n", searchChar, charCount);

} else {

printf("Character '%c' is not present in the string.\n", searchChar);

}

return 0;

}

2. 2 D integer array

Write a C program using functions and pointers for the following

1. Read and display n\*n matrix

2. Calculate the row total of a given matrix

3. Check whether the given matrix is identity matrix or not

#include <stdio.h>

#include <stdbool.h>

// Function to read an n x n matrix

void readMatrix(int \*matrix, int n) {

printf("Enter the elements of the %dx%d matrix:\n", n, n);

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

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

scanf("%d", (matrix + i \* n + j));

}

}

}

// Function to display an n x n matrix

void displayMatrix(const int \*matrix, int n) {

printf("Matrix:\n");

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

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

printf("%d ", \*(matrix + i \* n + j));

}

printf("\n");

}

}

// Function to calculate the row total of a matrix

void calculateRowTotal(const int \*matrix, int n, int row) {

int total = 0;

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

total += \*(matrix + row \* n + j);

}

printf("Row %d Total: %d\n", row, total);

}

// Function to check if the given matrix is an identity matrix

bool isIdentityMatrix(const int \*matrix, int n) {

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

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

if (i == j) {

if (\*(matrix + i \* n + j) != 1) {

return false;

}

} else {

if (\*(matrix + i \* n + j) != 0) {

return false;

}

}

}

}

return true;

}

int main() {

int n;

printf("Enter the size of the square matrix (n): ");

scanf("%d", &n);

int matrix[n][n];

// Read and display the matrix

readMatrix((int \*)matrix, n);

displayMatrix((int \*)matrix, n);

// Calculate row totals

int rowToCalculate;

printf("Enter the row number to calculate its total: ");

scanf("%d", &rowToCalculate);

calculateRowTotal((int \*)matrix, n, rowToCalculate);

// Check if the matrix is an identity matrix

if (isIdentityMatrix((int \*)matrix, n)) {

printf("The given matrix is an identity matrix.\n");

} else {

printf("The given matrix is not an identity matrix.\n");

}

return 0;

}

3. Using Structures in C , Create a Railway Reservation database storing the passenger

no as integer, passenger name which holds maximum of 25 characters, ticket price as

float for 5 passengers. Write a menu driven program to do the following: a)Read and

display the structure details. b)Search the given  passenger number using functions to

display as &quot;Passenger found&quot; or &quot;Passenger Not found&quot;. c)Modify the ticket price as

Rs.300 if the ticket price falls between the range of Rs.250 to Rs 299 using functions.

#include <stdio.h>

#include <string.h>

// Structure to represent passenger information

struct Passenger {

int passengerNumber;

char passengerName[26];

float ticketPrice;

};

// Function to read details of n passengers

void readPassengerDetails(struct Passenger passengers[], int n) {

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

printf("Enter Passenger Number: ");

scanf("%d", &passengers[i].passengerNumber);

printf("Enter Passenger Name: ");

scanf(" %[^\n]s", passengers[i].passengerName);

printf("Enter Ticket Price: ");

scanf("%f", &passengers[i].ticketPrice);

}

}

// Function to display details of n passengers

void displayPassengerDetails(const struct Passenger passengers[], int n) {

printf("Passenger Details:\n");

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

printf("Passenger Number: %d\n", passengers[i].passengerNumber);

printf("Passenger Name: %s\n", passengers[i].passengerName);

printf("Ticket Price: Rs. %.2f\n", passengers[i].ticketPrice);

printf("\n");

}

}

// Function to search for a passenger by passenger number

void searchPassenger(const struct Passenger passengers[], int n, int passengerNumber) {

int found = 0;

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

if (passengers[i].passengerNumber == passengerNumber) {

printf("Passenger found!\n");

found = 1;

break;

}

}

if (!found) {

printf("Passenger Not found.\n");

}

}

// Function to modify ticket prices within a specific range

void modifyTicketPrice(struct Passenger passengers[], int n) {

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

if (passengers[i].ticketPrice >= 250.0 && passengers[i].ticketPrice < 299.0) {

passengers[i].ticketPrice = 300.0;

}

}

}

int main() {

struct Passenger passengers[5]; // Assuming a maximum of 5 passengers

int choice, passengerNumber;

while (1) {

printf("Railway Reservation Database Menu:\n");

printf("1. Read Passenger Details\n");

printf("2. Display Passenger Details\n");

printf("3. Search Passenger by Number\n");

printf("4. Modify Ticket Prices\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

readPassengerDetails(passengers, 5);

break;

case 2:

displayPassengerDetails(passengers, 5);

break;

case 3:

printf("Enter Passenger Number to search: ");

scanf("%d", &passengerNumber);

searchPassenger(passengers, 5, passengerNumber);

break;

case 4:

modifyTicketPrice(passengers, 5);

printf("Ticket prices modified.\n");

break;

case 5:

return 0;

default:

printf("Invalid choice. Please try again.\n");

}

}

return 0;

}