**Q.1: Write a program to determine the maximum of 3 numbers.**

Input:

#include<stdio.h>

void main()

{

int a=20,b=40,c=60;

if(a>b && b>c)

{

printf("\n A is bigger");

}

else if (b>a && b>c)

{

printf("\n B is bigger");

}

else

{

printf("\n C is bigger");

}

}

Output:

C is bigger.

**Q.2:Write a program to swap the values of two variables.**

Input:

#include <stdio.h>

int main() {

int a, b, temp;

printf("Enter the first number (a): ");

scanf("%d", &a);

printf("Enter the second number (b): ");

scanf("%d", &b);

printf("Before swapping: a = %d, b = %d\n", a, b);

temp = a;

a = b;

b = temp;

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

return 0;

}

Output:

Enter the first number (a): 20

Enter the second number (b): 30

Before swapping: a = 20, b = 30

After swapping: a = 30, b = 20

**Q.3 Write a program that reads the percentage obtained by the students and determines and print the class obtained by the student as per the following rules**

**percentage class**

**0-39 Fail**

**40-59 Second Class**

**60-79 First Class**

**80-100 Distinction**

Input:

#include <stdio.h>

int main() {

float percentage;

printf("Enter the percentage obtained by the student: ");

scanf("%f", &percentage);

if (percentage >= 0 && percentage < 40) {

printf("Class obtained: Fail\n");

} else if (percentage >= 40 && percentage < 60) {

printf("Class obtained: Second class\n");

} else if (percentage >= 60 && percentage < 80) {

printf("Class obtained: First class\n");

} else if (percentage >= 80 && percentage <= 100) {

printf("Class obtained: Distinction\n");

} else {

printf("Invalid percentage. Please enter a value between 0 and 100.\n");

}

return 0;

}

output:

Enter the percentage obtained by the student: 90

Class obtained: Distinction

**Q.4 Write a program to calculate the area of circle/rectangle/triangle.**

**C indicate circle, R indicate rectangle, T indicate triangle.**

Input:

#include <stdio.h>

#define PI 3.14159265358979323846

int main() {

char shape;

float area;

printf("Enter the shape (C for Circle, R for Rectangle, T for Triangle): ");

scanf(" %c", &shape);

switch (shape) {

case 'C': case 'c': {

float radius;

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

scanf("%f", &radius);

area = PI \* radius \* radius;

printf("Area of the circle: %.2f\n", area);

break; }

case 'R': case 'r': {

float length, width;

printf("Enter the length of the rectangle: ");

scanf("%f", &length);

printf("Enter the width of the rectangle: ");

scanf("%f", &width);

area = length \* width;

printf("Area of the rectangle: %.2f\n", area);

break; }

case 'T': case 't': {

float base, height;

printf("Enter the base of the triangle: ");

scanf("%f", &base);

printf("Enter the height of the triangle: ");

scanf("%f", &height);

area = 0.5 \* base \* height;

printf("Area of the triangle: %.2f\n", area);

break; }

default:

printf("Invalid choice. Please enter C, R, or T.\n");

} }

Output:

Enter the shape (C for Circle, R for Rectangle, T for Triangle): R

Enter the length of the rectangle: 5

Enter the width of the rectangle: 6

Area of the rectangle: 30.00

**Q.5 Write a program that accept basic,HRA,and DA from the user and calculate total salary.**

INPUT:

#include <stdio.h>

int main() {

float basic, hra, da, totalSalary;

printf("Enter the basic salary: ");

scanf("%f", &basic);

printf("Enter the House Rent Allowance (HRA): ");

scanf("%f", &hra);

printf("Enter the Dearness Allowance (DA): ");

scanf("%f", &da);

totalSalary = basic + hra + da;

printf("Total Salary = %.2f\n", totalSalary);

return 0;

}

Output:

Enter the basic salary: 40000

Enter the House Rent Allowance (HRA): 500

Enter the Dearness Allowance (DA): 2000

Total Salary = 42500.00

**Q.6 Write a program to print the multiplication table of given number.**

Input:

#include <stdio.h>

int i;

int main() {

int number;

printf("Enter a number to print its multiplication table: ");

scanf("%d", &number);

printf("Multiplication table for %d:\n", number);

for ( i = 1; i <= 10; i++) {

printf("%d x %d = %d\n", number, i, number \* i);

}

return 0;

}

Output:

Enter a number to print its multiplication table: 5

Multiplication table for 5:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

**Q.7 Write a program to determine given number is prime or not.**

Input:

#include <stdio.h>

#include <stdbool.h>

int i;

bool isPrime(int num) {

if (num <= 1) return false;

if (num == 2 || num == 3) return true;

if (num % 2 == 0 || num % 3 == 0) return false;

for ( i = 5; i \* i <= num; i += 6) {

if (num % i == 0 || num % (i + 2) == 0) return false;

}

return true;

}

int main() {

int number;

printf("Enter a number: ");

scanf("%d", &number);

if (isPrime(number)) {

printf("%d is a prime number.\n", number);

} else {

printf("%d is not a prime number.\n", number);

}

return 0;

}

Output:

Enter a number: 56

56 is not a prime number.

**Q.8 Write a program to reverse a given number and display the sum of all digits.**

Input:

#include <stdio.h>

int main() {

int num, reversed = 0, original, digit, sum = 0;

printf("Enter an integer: ");

scanf("%d", &num);

original = num;

while (num != 0) {

digit = num % 10;

reversed = reversed \* 10 + digit;

sum += digit;

num /= 10;

}

printf("Reversed Number: %d\n", reversed);

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

return 0;

}

Output:

Enter an integer: 65

Reversed Number: 56

Sum of Digits: 11

**Q.9 Write a program to accept two numbers and perform basic operatons of calculator (+,-,\*,/)(Use switch case).**

Input:

#include <stdio.h>

int main() {

float num1, num2;

char operator;

float result;

printf("Enter the first number: ");

scanf("%f", &num1);

printf("Enter the second number: ");

scanf("%f", &num2);

printf("Enter an operator (+, -, \*, /): ");

scanf(" %c", &operator);

switch (operator) {

case '+':

result = num1 + num2;

printf("Result: %.2f\n", result);

break;

case '-':

result = num1 - num2;

printf("Result: %.2f\n", result);

break;

case '\*':

result = num1 \* num2;

printf("Result: %.2f\n", result);

break;

case '/':

if (num2 != 0) {

result = num1 / num2;

printf("Result: %.2f\n", result);

} else {

printf("Error: Division by zero is not allowed.\n");}

}

}

output:

Enter the first number: 5

Enter the second number: 6

Enter an operator (+, -, \*, /): +

Result: 11.00

**Q.10 Write a program to find maximum and minimum element from 1-Dimensional array.**

Input:

#include <stdio.h>

int main() {

int n, i;

int max, min;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d elements:\n", n);

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

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

}

max = min = arr[0];

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

if (arr[i] > max) {

max = arr[i];

}

if (arr[i] < min) {

min = arr[i];

}

}

printf("Maximum element: %d\n", max);

printf("Minimum element: %d\n", min);

return 0;

}

Output:

Enter the number of elements in the array: 3

Enter 3 elements:

5 6 7

Maximum element: 7

Minimum element: 5

**Q.11 Write a program to sort given array in ascending order.**

Input:

#include <stdio.h>

void bubbleSort(int arr[], int n) {

int i, j, temp;

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;

} } } }

int main() {

int n, i;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d elements:\n", n);

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

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

}

bubbleSort(arr, n);

printf("Sorted array in ascending order:\n");

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

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

}

printf("\n");

return 0;

}

Output:

Enter the number of elements in the array: 2

Enter 2 elements:

5

6

Sorted array in ascending order:

5 6

**Q.12 Write a program to add two matrices.**

INPUT:

#include <stdio.h>

int main()

{ int rows, cols;

printf("Enter the number of rows: ");

scanf("%d", &rows);

printf("Enter the number of columns: ");

scanf("%d", &cols);

int A[rows][cols], B[rows][cols], C[rows][cols];

printf("Enter elements of matrix A :\n");

int i,j;

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

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

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

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

printf("Enter elements of matrix B :\n");

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

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

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

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

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

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

C[i][j] = A[i][j] + B[i][j];

} }

printf("Resulting matrix C (A + B):\n");

for (i = 0; i < rows; i++)

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

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

printf("\n"); }

return 0; }

Output:

Enter the number of rows: 2

Enter the number of columns: 2

Enter elements of matrix A :

A [0][0]: 2

A [0][1]: 3

A [1][0]: 1

A [1][1]: 3

Enter elements of matrix B :

B [0][0]: 1

B [0][1]: 2

B [1][0]: 2

B [1][1]: 3

Resulting matrix C (A + B):

3 5

3 6

**Q.13 Write a program to find element at given position from 2-Dimensional array.**

Input: #include <stdio.h>

#define MAX\_ROWS 10

#define MAX\_COLS 10

int main() { int i; int j;

int rows, cols;

int array[MAX\_ROWS][MAX\_COLS];

printf("Enter the number of rows (max %d): ", MAX\_ROWS);

scanf("%d", &rows);

printf("Enter the number of columns (max %d): ", MAX\_COLS);

scanf("%d", &cols);

printf("Enter elements of the 2D array:\n");

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

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

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

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

int rowPos, colPos;

printf("Enter the row index (0 to %d): ", rows - 1);

scanf("%d", &rowPos);

printf("Enter the column index (0 to %d): ", cols - 1);

scanf("%d", &colPos);

if (rowPos >= 0 && rowPos < rows && colPos >= 0 && colPos < cols) {

printf("Element at position [%d][%d]: %d\n", rowPos, colPos, array[rowPos][colPos]);

} else

{ printf("Error: Index out of bounds!\n"); }

return 0;

}

Output:

Enter the number of rows (max 10): 2

Enter the number of columns (max 10): 3

Enter elements of the 2D array:

Element [0][0]: 1

Element [0][1]: 2

Element [0][2]: 2

Element [1][0]: 3

Element [1][1]: 5

Element [1][2]: 6

Enter the row index (0 to 1): 1

Enter the column index (0 to 2): 2

Element at position [1][2]: 6

**Q.14 Write a program that will read a text and count all occurances of a particular character using function.**

INPUT:

#include <stdio.h>

int countCharacterOccurrences(char text[],

char ch ) {

int i;

int count = 0;

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

if (text[i] == ch) {

count++;

}

}

return count;

}

int main() {

char text[100];

char ch;

printf("Enter a text : ");

scanf("%99s",text);

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

scanf(" %c", &ch);

int occurrences = countCharacterOccurrences(text, ch);

printf("The character '%c' occurs %d times in the text.\n", ch, occurrences);

return 0;

}

OUTPUT:

Enter a text : krimisha

Enter the character to count: i

The character 'i' occurs 2 times in the text.

**Q.15 Write a function which returns 1 if the given number is palindrome otherwise returns 0.**

INPUT:

#include <stdio.h>

int is Palindrome(int num) {

int original = num;

int reversed = 0;

while (num > 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

return (original == reversed) ? 1 : 0;

}

int main() {

int num;

printf("Enter a number to check if it's a palindrome: ");

scanf("%d", &num);

if (isPalindrome(num)) {

printf("1 (the number is a palindrome)\n");

} else {

printf("0 (the number is not a palindrome.\n");

}

return 0;

}

OUTPUT:

Enter a number to check if it's a palindrome:

232

1 (the number is a palindrome)

**Q.16 Write a recursive function for finding the factorial of a number.**

Input:

#include <stdio.h>

int factorial(int n) {

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

return 1;

}

return n \* factorial(n - 1);

}

int main() {

int number;

printf("Enter a number: ");

scanf("%d", &number);

if (number < 0) {

printf("Factorial is not defined for negative numbers.\n");

} else {

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

}

return 0;

}

Output:

Enter a number: 5

Factorial of 5 is 120

**Q. 17 Write a program to perform summation of all elements of array using pointer.**

INPUT:

#include <stdio.h>

int main() {

int arr[100], n, i;

int \*ptr = arr;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

scanf("%d", ptr + i);

}

int sum = 0;

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

sum += \*(ptr + i);

}

printf("The sum of the elements is: %d\n", sum);

return 0;

}

OUTPUT:

Enter the number of elements in the array: 3

Enter 3 elements:

5 6 7

The sum of the elements is: 18

**Q.18 Write a function using pointers to exchange the value stored in two locations in the memory.**

INPUT:

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

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

scanf("%d %d", &x, &y);

printf("Before swapping: x = %d, y = %d\n", x, y);

swap(&x, &y);

printf("After swapping: x = %d, y = %d\n", x, y);

return 0;

}

OUTPUT:

Enter two integers:

4 6

Before swapping: x = 4, y = 6

After swapping: x = 6, y = 4

**Q.19 Write a program to create structure Student with student’s roll no, name and marks of three subjects(Maths, Science and English) and display the details of student with total marks of all subjects along with the percentage and passing class in proper format.**

INPUT:

#include <stdio.h>

struct Student {

int rollNo;

char name[50];

float marks[3];

};

void displayStudentDetails(struct Student student) {

float total = 0.0;

int i;

for ( i = 0; i < 3; i++) {

total += student.marks[i];

}

float percentage = (total / 300) \* 100;

char \*passingClass;

if (percentage >= 60) {

passingClass = "First Class";

} else if (percentage >= 50) {

passingClass = "Second Class";

} else if (percentage >= 40) {

passingClass = "Third Class";

} else {

passingClass = "Fail";

}

printf("\n--- Student Details ---\n");

printf("Roll No: %d\n", student.rollNo);

printf("Name: %s\n", student.name);

printf("Marks:\n");

printf(" Maths: %.2f\n", student.marks[0]);

printf(" Science: %.2f\n", student.marks[1]);

printf(" English: %.2f\n", student.marks[2]);

printf("Total Marks: %.2f\n", total);

printf("Percentage: %.2f%%\n", percentage);

printf("Passing Class: %s\n", passingClass);

}

int main() {

struct Student student;

printf("Enter Roll No: ");

scanf("%d", &student.rollNo);

printf("Enter Name: ");

scanf(" %[^\n]", student.name);

printf("Enter marks for Maths, Science, and English: ");

scanf("%f %f %f", &student.marks[0], &student.marks[1], &student.marks[2]);

displayStudentDetails(student);

return 0;

}

OUTPUT:

Enter Roll No: 5

Enter Name: KRIMISHA

Enter marks for Maths, Science, and English: 90 98 96

--- Student Details ---

Roll No: 5

Name: KRIMISHA

Marks:

Maths: 90.00

Science: 98.00

English: 96.00

Total Marks: 284.00

Percentage: 94.67%

Passing Class: First Class

**Q.20 Write a program to create structure Time (data members: int h, int m, int sec).Read a value as seconds from user to display new time after adding seconds to Time structure.**

INPUT:

#include <stdio.h>

struct Time {

int h;

int m;

int sec;

};

void addSeconds(struct Time \*time, int secondsToAdd) {

time->sec += secondsToAdd;

time->m += time->sec / 60;

time->sec = time->sec % 60;

time->h += time->m / 60;

time->m = time->m % 60;

time->h = time->h % 24;

}

void displayTime(struct Time time) {

printf("Updated Time: %02d:%02d:%02d\n", time.h, time.m, time.sec);

}

int main() {

struct Time time;

printf("Enter time in HH MM SS format: ");

scanf("%d %d %d", &time.h, &time.m, &time.sec);

int secondsToAdd;

printf("Enter seconds to add: ");

scanf("%d", &secondsToAdd);

addSeconds(&time, secondsToAdd);

displayTime(time);

return 0;

}

OUTPUT:

Enter time in HH MM SS format: 10 02 55

Enter seconds to add: 2

Updated Time: 10:02:57

**Q.21 Write a program to define a structure called book. Write a program to read information about 5 books and display books details in ascending order of price in proper format.**

INPUT: #include <stdio.h>

#include <string.h>

#define MAX\_BOOKS 5

struct Book { char title[50];

char author[50];

float price; };

void sortBooksByPrice(struct Book books[], int n) {

struct Book temp;

int i; int j;

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

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

if (books[j].price > books[j + 1].price) {

temp = books[j];

books[j] = books[j + 1];

books[j + 1] = temp; } } } }

int main() {

struct Book books[MAX\_BOOKS];

int i;

for ( i = 0; i < MAX\_BOOKS; i++) {

printf("Enter details for book %d:\n", i + 1);

printf("Title: ");

scanf(" %[^\n]%\*c", books[i].title);

printf("Author: ");

scanf(" %[^\n]%\*c", books[i].author);

printf("Price: ");

scanf("%f", &books[i].price); }

sortBooksByPrice(books, MAX\_BOOKS);

printf("\nBooks in ascending order of price:\n");

printf("%-30s %-30s %-10s\n", "Title", "Author", "Price");

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

for (i = 0; i < MAX\_BOOKS; i++) {

printf("%-30s %-30s $%-9.2f\n", books[i].title, books[i].author, books[i].price) }

return 0;

}

OUTPUT:

Enter details for book 1:

Title: The Night Circus

Author: Erin Morgenstern

Price: 5000

Enter details for book 2:

Title: Educated

Author: Tara Westover

Price: 3500

Enter details for book 3:

Title: The Alchemist

Author: Paulo Coelho

Price: 2550

Enter details for book 4:

Title: Where the Crawdads Sing

Author: Delia Owens

Price: 3550

Enter details for book 5:

Title: 1984

Author: George Orwell

Price: 4550

Books in ascending order of price:

Title Author Price

---------------------------------------------

The Alchemist Paulo Coelho $2550.00

Educated Tara Westover $3500.00

Where the Crawdads Sing Delia Owens $3550.00

1984 George Orwell $4550.00

The Night Circus Erin Morgenstern $5000.00

**Q-22. Write a program to copy the contents of one file to another and also print the no. of lines in the first file.**

CODE:

#include <stdio.h>

#include <stdlib.h>

void main(){

int nline;

char c;

FILE \*f1, \*f2;

f1 = fopen("txt1.txt", "r");

f2 = fopen("txt2.txt", "w");

if(!f1){

printf("Something went wrong while opening the file one");

exit(1);

}

if(!f2){

printf("Something went wrong whie opening the file two");

exit(1);

}

c = fgetc(f1);

while(c != EOF){

if(c == '\n') nline++;

fputc(c, f2);

c = fgetc(f1);

}

printf("Number of line in the file one is : %d", nline);

}

**Q-23. Write a function to read a file and count the no. of characters, spaces, newlines and no. of words in a given text file.**

CODE:

include <stdio.h>

int main(){

int cc = 0, sc = 0, nlc = 0, wc = 0;

char c;

FILE \*f1 = fopen("txt1.txt", "r");

if(!f1){

printf("Something went wrong while opening file!");

return 0;

}

c = fgetc(f1);

while(c != EOF){

if(c == 10) nlc++;

if(c == ' '){

sc++;

wc++;

};

cc++;

c = fgetc(f1);

}

nlc++;

if(c) wc++;

printf("Charcter : %d\n", cc);

printf("Words : %d\n", wc);

printf("Spaces : %d\n", sc);

printf("Lines : %d\n", nlc);

return 1;

}

**Q-24. Write an interactive menu driven program that will access the data file created in the above problem to do one of the following tasks:**

**a. Determine the telephone number of a specific customers.**

**b. Determine the customer whose telephone no. is specified.**

**c. Add a new record.**

**d. Delete a record**

**e. Generate the listing of all the customers and their telephone numbers**

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Customer {

char name[20];

char telephone[10];

};

struct Customer \*custs = NULL;

struct Customer s;

FILE \*file;

int n = 0;

void printRecord(struct Customer record) {

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

printf("Name: %s\n", record.name);

printf("Telephone Number: %s\n", record.telephone);

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

}

void printall(){

int i;

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

printRecord(custs[i]);

}

}

void extractData() {

while (fread(&s, sizeof(s), 1, file) == 1) {

struct Customer \*temp = realloc(custs, (n + 1) \* sizeof(struct Customer));

if (temp == NULL) {

printf("Memory allocation failed!\n");

exit(1);

}

custs = temp;

custs[n++] = s;

}

printf("\nData extracted from the file.\n");

}

void openFile() {

char filename[50];

printf("Enter the name of the file to open or create: ");

fgets(filename, sizeof(filename), stdin);

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

file = fopen(filename, "a+b");

if (!file) {

printf("Could not open file!\n");

return;

}

printf("File opened successfully.\n");

extractData();

}

void writeBinary() {

int i;

if (file) {

fseek(file, 0, SEEK\_SET);

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

fwrite(&custs[i], sizeof(struct Customer), 1, file);

}

printf("\nSuccessfully saved data to the file.\n");

} else {

printf("File is not opened!\n");

}

}

void inputRecord() {

printf("\nEnter the details of the new customer:\n");

printf("Enter the name: ");

fgets(s.name, sizeof(s.name), stdin);

s.name[strcspn(s.name, "\n")] = 0;

printf("Enter the telephone number: ");

fgets(s.telephone, sizeof(s.telephone), stdin);

s.telephone[strcspn(s.telephone, "\n")] = 0;

struct Customer \*temp = realloc(custs, (n + 1) \* sizeof(struct Customer));

if (temp == NULL) {

printf("Memory allocation failed!\n");

exit(1);

}

custs = temp;

custs[n++] = s;

printf("\nRecord added to the list.\n");

}

void deleteRecord() {

int i, j;

char str[20];

printf("Enter the mobile number or the name of the customer to delete: ");

fgets(str, sizeof(str), stdin);

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

int found = 0;

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

if (strcmp(custs[i].name, str) == 0 || strcmp(custs[i].telephone, str) == 0) {

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

custs[j] = custs[j + 1];

}

n--;

struct Customer \*temp = realloc(custs, n \* sizeof(struct Customer));

if (temp == NULL && n > 0) {

printf("Memory allocation failed!\n");

exit(1);

}

custs = temp;

found = 1;

printf("Record deleted.\n");

break;

}

}

if (!found) {

printf("No such record found to delete.\n");

}

}

void getRecord() {

int i;

char str[20];

printf("\nEnter the telephone number or name: ");

fgets(str, sizeof(str), stdin);

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

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

if (strcmp(custs[i].name, str) == 0 || strcmp(custs[i].telephone, str) == 0) {

printRecord(custs[i]);

getchar();

return;

}

}

printf("No such record found.\n");

getchar();

}

void menu() {

if(!file) openFile();

char c;

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

printf("\nOpen or create file (o)\n");

printf("Add new record (a)\n");

printf("Get record (g)\n");

printf("Delete customer (d)\n");

printf("Save changes (s)\n");

printf("Print all record (l)\n");

printf("Exit (x)\n");

printf("Input: ");

scanf(" %c", &c);

while (getchar() != '\n');

switch (c) {

case 'a': inputRecord(); break;

case 'g': getRecord(); break;

case 'd': deleteRecord(); break;

case 's': writeBinary(); break;

case 'o': openFile(); break;

case 'l': printall(); break;

case 'x':

if (file) fclose(file);

free(custs);

exit(0);

default: printf("\nInvalid input!!!\n");

}

menu();

}

int main() {

menu();

free(custs);

return 0;

}

Q-25. Use a structure of Employee to write records of employee to a file. Include a menu that will allow the user to select any of the following features

a. Add a new record.

b. Delete a record.

c. Modify an existing record.

d. Retrieve and display an entire record for a givenID/Name.

e. Generate a complete list of all employee names,addresses and telephone numbers.

f. End of the computation/Exit.

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_EMPLOYEES 100

#define FILENAME "employees.dat"

struct Employee {

int id;

char name[50];

char address[100];

char phone[15];

};

void addRecord();

void deleteRecord();

void modifyRecord();

void retrieveRecord();

void listRecords();

void clearBuffer();

int main() {

int choice;

do {

printf("\nEmployee Management System\n");

printf("1. Add a new record\n");

printf("2. Delete a record\n");

printf("3. Modify an existing record\n");

printf("4. Retrieve and display a record\n");

printf("5. Generate a complete list of employees\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

clearBuffer();

switch (choice) {

case 1: addRecord(); break;

case 2: deleteRecord(); break;

case 3: modifyRecord(); break;

case 4: retrieveRecord(); break;

case 5: listRecords(); break;

case 6: printf("Exiting...\n"); break;

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

}

} while (choice != 6);

return 0;

}

void clearBuffer() {

while (getchar() != '\n');

}

void addRecord() {

struct Employee emp;

FILE \*file = fopen(FILENAME, "ab");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("Enter ID: ");

scanf("%d", &emp.id);

clearBuffer();

printf("Enter Name: ");

fgets(emp.name, sizeof(emp.name), stdin);

strtok(emp.name, "\n");

printf("Enter Address: ");

fgets(emp.address, sizeof(emp.address), stdin);

strtok(emp.address, "\n");

printf("Enter Phone: ");

fgets(emp.phone, sizeof(emp.phone), stdin);

strtok(emp.phone, "\n");

fwrite(&emp, sizeof(struct Employee), 1, file);

fclose(file);

printf("Record added successfully.\n");

}

void deleteRecord() {

int id;

struct Employee emp;

FILE \*file = fopen(FILENAME, "rb");

FILE \*tempFile = fopen("temp.dat", "wb");

if (file == NULL || tempFile == NULL) {

printf("Error opening file!\n");

return;

}

printf("Enter ID of the employee to delete: ");

scanf("%d", &id);

int found = 0;

while (fread(&emp, sizeof(struct Employee), 1, file)) {

if (emp.id != id) {

fwrite(&emp, sizeof(struct Employee), 1, tempFile);

} else {

found = 1;

}

}

fclose(file);

fclose(tempFile);

remove(FILENAME);

rename("temp.dat", FILENAME);

if (found) {

printf("Record deleted successfully.\n");

} else {

printf("Record with ID %d not found.\n", id);

}

}

void modifyRecord() {

int id;

struct Employee emp;

FILE \*file = fopen(FILENAME, "r+b");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("Enter ID of the employee to modify: ");

scanf("%d", &id);

int found = 0;

while (fread(&emp, sizeof(struct Employee), 1, file)) {

if (emp.id == id) {

found = 1;

printf("Enter new Name: ");

clearBuffer();

fgets(emp.name, sizeof(emp.name), stdin);

strtok(emp.name, "\n"); // Remove newline character

printf("Enter new Address: ");

fgets(emp.address, sizeof(emp.address), stdin);

strtok(emp.address, "\n"); // Remove newline character

printf("Enter new Phone: ");

fgets(emp.phone, sizeof(emp.phone), stdin);

strtok(emp.phone, "\n"); // Remove newline character

fseek(file, -sizeof(struct Employee), SEEK\_CUR); // Move to the record's position

fwrite(&emp, sizeof(struct Employee), 1, file);

break;

}

}

fclose(file);

if (found) {

printf("Record modified successfully.\n");

} else {

printf("Record with ID %d not found.\n", id);

}

}

void retrieveRecord() {

int id;

struct Employee emp;

FILE \*file = fopen(FILENAME, "rb");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("Enter ID of the employee to retrieve: ");

scanf("%d", &id);

int found = 0;

while (fread(&emp, sizeof(struct Employee), 1, file)) {

if (emp.id == id) {

found = 1;

printf("ID: %d\n", emp.id);

printf("Name: %s\n", emp.name);

printf("Address: %s\n", emp.address);

printf("Phone: %s\n", emp.phone);

break;

}

}

fclose(file);

if (!found) {

printf("Record with ID %d not found.\n", id);

}

}

void listRecords() {

struct Employee emp;

FILE \*file = fopen(FILENAME, "rb");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

printf("\nList of Employees:\n");

while (fread(&emp, sizeof(struct Employee), 1, file)) {

printf("ID: %d, Name: %s, Address: %s, Phone: %s\n", emp.id, emp.name, emp.address, emp.phone);

}

fclose(file);

}