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

CODE:

#include <stdio.h>

int main() {

int a, b, c;

printf("Enter three numbers:\n");

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

if (a >= b && a >= c) {

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

} else if (b >= a && b >= c) {

printf("The maximum number is %d.\n", b);

} else {

printf("The maximum number is %d.\n", c);

}

}

OUTPUT:

Enter three numbers:

23

65

52

The maximum number is 65.

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

CODE:

#include <stdio.h>

int main() {

int a, b, temp;

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

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

temp = a;

a = b;

b = temp;

printf("After swapping: \n");

printf("a = %d\n", a);

printf("b = %d\n", b);

}

OUTPUT:

Enter two numbers:

20

60

After swapping:

a = 60

b = 20

Q-3: Write a program that reads the percentage obtained by the students and determines and

prints 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

CODE:

#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: 60

Class obtained: First class

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

C indicate circle ,

R indicate rectangle,

T indicate triangle.

use symbolic constant to define the value of pie.

CODE:

#include <stdio.h>

int main() {

char shape;

float area, length, width, radius, base, height, PI=3.14159;

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

scanf(" %c", &shape)

if (shape == 'C' || shape == 'c')

{

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

scanf("%f", &radius);

area = PI \* radius \* radius;

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

}

else if (shape == 'R' || shape == 'r')

{

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

scanf("%f %f", &length, &width);

area = length \* width;

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

}

else if (shape == 'T' || shape == 't')

{

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

scanf("%f %f", &base, &height);

area = 0.5 \* base \* height;

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

}

else

{

printf("Invalid shape type entered.\n");

}

}

OUTPUT:

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

Enter the base and height of the triangle: 5.9

6.9

Area of the triangle: 20.36

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

CODE:

#include <stdio.h>

int main()

{

float basic, hra, da, total\_salary;

printf("Enter Basic Salary: ");

scanf("%f", &basic);

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

scanf("%f", &hra);

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

scanf("%f", &da);

total\_salary = basic + hra + da;

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

return 0;

}

OUTPUT:

Enter Basic Salary: 50000

Enter House Rent Allowance (HRA): 10

Enter Dearness Allowance (DA): 600

Total Salary = 50610.00

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

CODE:

#include <stdio.h>

int main() {

int num;

int i;

printf("Enter the number for which you want to print the multiplication table: ");

scanf("%d", &num);

printf("Multiplication table of %d:\n", num);

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

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

}

return 0;

}

OUTPUT:

Enter the number for which you want to print the multiplication table: 6

Multiplication table of 6:

6 x 1 = 6

6 x 2 = 12

6 x 3 = 18

6 x 4 = 24

6 x 5 = 30

6 x 6 = 36

6 x 7 = 42

6 x 8 = 48

6 x 9 = 54

6 x 10 = 60

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

CODE:

#include <stdio.h>

int main() {

int num, i, isPrime = 1;

printf("Enter a number: ");

scanf("%d", &num);

if (num <= 1) {

isPrime = 0;

} else {

for (i = 2; i \* i <= num; i++) {

if (num % i == 0) {

isPrime = 0;

break;

}

}

}

if (isPrime) {

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

}

else {

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

}

return 0;

}

OUTPUT:

Enter a number: 53

53 is a prime number.

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

CODE:

#include <stdio.h>

int reverseNumber(int num) {

int reversed = 0;

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

return reversed;

}

int sumOfDigits(int num) {

int sum = 0;

while (num != 0) {

sum += num % 10;

num /= 10;

}

return sum;

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

int reversed = reverseNumber(num);

int sum = sumOfDigits(num);

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

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

return 0;

}

OUTPUT:

Enter a number: 598

Reversed number: 895

Sum of digits: 22

Q-9: Write a program to accept two numbers and perform basic operation of calculator

(+,-,\*,/).(Use switch...case)

CODE:

#include <stdio.h>

int main() {

float num1, num2;

char operator;

printf("Enter first number: ");

scanf("%f", &num1);

printf("Enter second number: ");

scanf("%f", &num2);

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

scanf(" %c", &operator);

switch (operator) {

case '+':

printf("%.2f + %.2f = %.2f\n", num1, num2, num1 + num2);

break;

case '-':

printf("%.2f - %.2f = %.2f\n", num1, num2, num1 - num2);

break;

case '\*':

printf("%.2f \* %.2f = %.2f\n", num1, num2, num1 \* num2);

break;

case '/':

if (num2 != 0) {

printf("%.2f / %.2f = %.2f\n", num1, num2, num1 / num2);

} else {

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

}

break;

default:

printf("Error: Invalid operator.\n");

break;

}

return 0;

}

OUTPUT:

Enter first number: 20

Enter second number: 90

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

20.00 \* 90.00 = 1800.00

Q-10: Write a program to find maximum and minimum element from

1-Dimensional array.

CODE:

#include<stdio.h>

int main()

{

int a[100], i, size,max,min;

printf("\n enter the size of arr : ");

scanf("%d",&size);

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

{

printf("\n enter the element of arr %d : ", i);

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

}

max=a[0];

min=a[0];

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

{

if(max<a[i])

{

max=a[i];

}

if(min>a[i])

{

min=a[i];

}

}

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

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

}

OUTPUT:

enter the size of arr : 3

enter the element of arr 0 : 95

enter the element of arr 1 : 85

enter the element of arr 2 : 51

maximum element is : 95

minimum element is : 51

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

CODE:

#include <stdio.h>

int main() {

int n, i, j;

int temp;

int arr[n];

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

scanf("%d", &n);

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

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

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

}

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;

}

}

}

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

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

{

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

}

printf("\n");

return 0;

}

OUTPUT:

Enter the number of elements in the array: 6

Enter 6 elements: 92

60

35

84

99

96

Sorted array in ascending order: 35 60 84 92 96 99

Q-12: Write a program to add two matrices.

CODE:

#include<stdio.h>

int main()

{

int a[100][100], b[100][100], r, c, i, j, sum[100][100];

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

scanf("%d",&r);

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

scanf("%d",&c);

printf("\n Enter the elements of 1st matrices :- ");

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

{

for(j=0;j<c;j++)

{

printf("\n Enter the element of a[%d %d] : ",i,j);

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

}

}

printf("\n Enter the elements of 2nd matrices :- ");

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

{

for(j=0;j<c;j++)

{

printf("\n Enter the element of a[%d %d] : ",i,j);

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

}

}

printf("\n The elements of 1st matrices is :- \n");

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

{

for(j=0;j<c;j++)

{

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

}

printf("\n");

}

printf("\n The elements of 2nd matrices is :- \n");

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

{

for(j=0;j<c;j++)

{

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

}

printf("\n");

}

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

{

for(j=0;j<c;j++)

{

sum[i][j] = a[i][j] + b[i][j];

}

}

printf("\n Sum of two matrices: \n");

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

{

for(j=0;j<c;j++)

{

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

}

printf("\n");

}

}

OUTPUT:

Enter the number of rows : 2

Enter the number of columns : 3

Enter the elements of 1st matrices :-

Enter the element of a[0 0] : 45

Enter the element of a[0 1] : 95

Enter the element of a[0 2] : 56

Enter the element of a[1 0] : 15

Enter the element of a[1 1] : 12

Enter the element of a[1 2] : 16

Enter the elements of 2nd matrices :-

Enter the element of a[0 0] : 46

Enter the element of a[0 1] : 15

Enter the element of a[0 2] : 95

Enter the element of a[1 0] : 85

Enter the element of a[1 1] : 62

Enter the element of a[1 2] : 52

The elements of 1st matrices is :-

45 95 56

15 12 16

The elements of 2nd matrices is :-

46 15 95

85 62 52

Sum of two matrices:

91 110 151

100 74 68

Q-13: Write a program to find element at given position from 2-

Dimensional array.

CODE:

#include<stdio.h>

int main()

{

int a[100][100], r, c, i, j, find, count=0;

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

scanf("%d",&r);

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

scanf("%d",&c);

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

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

{

for(j=0;j<c;j++)

{

printf("\n Enter the element of a[%d %d] : ",i,j);

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

}

}

printf("\n Enter the element you want to find position : ");

scanf("%d",&find);

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

{

for(j=0; j<c; j++)

{

if(a[i][j] == find)

{

printf("\n Item found at [%d, %d]", i, j);

count++;

}

}

}

if(count==0)

printf("\n Item Not found");

}

OUTPUT:

Enter the number of rows : 2

Enter the number of columns : 2

Enter the elements of matrices :-

Enter the element of a[0 0] : 50

Enter the element of a[0 1] : 60

Enter the element of a[1 0] : 40

Enter the element of a[1 1] : 90

Enter the element you want to find position : 40

Item found at [1, 0]

Q-14: Write a program that will read a text and count all occurrences

of a particular character using function.

CODE:

#include <stdio.h>

int countOccurrences(const char \*text, char target)

{

int count = 0;

while (\*text) {

if (\*text == target) {

count++;

}

text++;

}

return count;

}

int main() {

char text[200];

char target;

printf("Enter a text: ");

fgets(text, sizeof(text), stdin);

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

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

scanf(" %c", &target);

int count = countOccurrences(text, target);

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

return 0;

}

OUTPUT:

Enter a text: DDDGGL

Enter the character to count: G

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

Q-15: Write a function which returns 1 if the given number is

palindrome otherwise returns 0.

CODE:

#include <stdio.h>

int isPalindrome(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 number;

printf("Enter a number: ");

scanf("%d", &number);

if (isPalindrome(number)) {

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

} else {

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

}

return 0;

}

OUTPUT:

Enter a number: 2

2 is a palindrome.

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

CODE:

#include <stdio.h>

int factorial(int n)

{

if (n <= 1) {

return 1;

}

return n \* factorial(n - 1);

}

int main()

{

int number;

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number < 0) {

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

} else {

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

}

return 0;

}

OUTPUT:

Enter a positive integer: GHFDD

The factorial of 0 is 1.

Q-17. Write a program to perform summation of all elements of

array using pointers.

CODE:

#include <stdio.h>

int sumArray(int \*arr, int size) {

int i, sum = 0;

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

sum += \*(arr + i);

}

return sum;

}

int main() {

int i, size;

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

scanf("%d", &size);

int arr[size];

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

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

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

}

int total = sumArray(arr, size);

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

return 0;

}

OUTPUT:

Enter the number of elements in the array: 2

Enter 2 elements:

10

30

The sum of all elements in the array is: 40

Q-18. Write a function using pointers to exchange the value

stored in two locations in the memory.

CODE:

#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 to swap:\n");

printf("First value: ");

scanf("%d", &x);

printf("Second value: ");

scanf("%d", &y);

swap(&x, &y);

printf("After swapping:\n");

printf("First value: %d\n", x);

printf("Second value: %d\n", y);

return 0;

}

OUTPUT:

Enter two integers to swap:

First value: 10

Second value: 20

After swapping:

First value: 20

Second value: 10

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.

CODE:

#include <stdio.h>

#define SUBJECTS 3

struct Student {

int roll\_no;

char name[50];

float marks[SUBJECTS];

};

void calculateResults(struct Student \*student, float \*total, float \*percentage) {

int i;

\*total = 0;

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

\*total += student->marks[i];

}

\*percentage = (\*total / (SUBJECTS \* 100)) \* 100;

}

const char\* determineClass(float percentage) {

if (percentage >= 60) {

return "First Class";

} else if (percentage >= 50) {

return "Second Class";

} else if (percentage >= 40) {

return "Third Class";

} else {

return "Fail";

}

}

int main() {

struct Student student;

printf("Enter Roll Number: ");

scanf("%d", &student.roll\_no);

printf("Enter Name: ");

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

printf("Enter Marks for Maths: ");

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

printf("Enter Marks for Science: ");

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

printf("Enter Marks for English: ");

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

float total, percentage;

calculateResults(&student, &total, &percentage);

const char\* passing\_class = determineClass(percentage);

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

printf("Roll Number: %d\n", student.roll\_no);

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", passing\_class);

return 0;

}

OUTPUT:

Enter Roll Number: 10

Enter Name: divy

Enter Marks for Maths: 99

Enter Marks for Science: 97

Enter Marks for English: 95

--- Student Details ---

Roll Number: 10

Name: divy

Marks:

Maths: 99.00

Science: 97.00

English: 95.00

Total Marks: 291.00

Percentage: 97.00%

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.

CODE:

#include <stdio.h>

struct Time {

int h;

int m;

int sec;

};

void normalizeTime(struct Time \*t) {

if (t->sec >= 60) {

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

t->sec = t->sec % 60;

}

if (t->m >= 60) {

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

t->m = t->m % 60;

}

t->h = t->h % 24;

}

int main() {

struct Time currentTime;

int additionalSeconds;

printf("Enter current time (hours, minutes, seconds):\n");

printf("Hours: ");

scanf("%d", &currentTime.h);

printf("Minutes: ");

scanf("%d", &currentTime.m);

printf("Seconds: ");

scanf("%d", &currentTime.sec);

printf("Enter seconds to add: ");

scanf("%d", &additionalSeconds);

currentTime.sec += additionalSeconds;

normalizeTime(&currentTime);

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

return 0;

}

OUTPUT:

Enter current time (hours, minutes, seconds):

Hours: 10

Minutes: 20

Seconds: 60

Enter seconds to add: 8000

Updated Time: 12:34:20

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.

CODE:

#include <stdio.h>

#include <stdlib.h>

struct Book{

int id, releaseYear;

float price;

char title[20], description[50], author[15];

};

void printBook(struct Book book){

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

printf("Title : %s\n", book.title);

printf("Description : %s\n", book.description);

printf("Author : %s\n", book.author);

printf("Release year : %d\n", book.releaseYear);

printf("Price : %f", book.price);

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

}

void sort(struct Book \*bkPtr, int len){

int i, j;

struct Book temp;

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

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

if(bkPtr[j+1].price < bkPtr[j].price){

temp = \*(bkPtr + j);

\*(bkPtr + j) = \*(bkPtr + j + 1);

\*(bkPtr + j + 1) = temp;

}

}

}

}

int main(){

int n, i;

struct Book \*books;

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

scanf("%d", &n);

books = (struct Book \*)malloc(n \* sizeof(struct Book));

if(!books) return printf("memory allocation failed !");

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

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

printf("Enter the id of the book : ");

scanf("%d", &books[i].id);

printf("Enter the price of the book : ");

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

fflush(stdin);

printf("Enter the title of the book : ");

gets(books[i].title);

fflush(stdin);

printf("Enter the name of the author : ");

gets(books[i].author);

fflush(stdin);

printf("Enter the description of the book : ");

gets(books[i].description);

fflush(stdin);

printf("Enter the release year of the book : ");

scanf("%d", &books[i].releaseYear);

}

sort(&books[0], n);

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

printBook(books[i]);

}

return 0;

}

OUTPUT:

Enter the number of books : 2

Enter the details of the book 1

Enter the id of the book : 10

Enter the price of the book : 900

Enter the title of the book : dsa

Enter the name of the author : divya

Enter the release year of the book : 1999

Enter the details of the book 2

Enter the id of the book : 22

Enter the price of the book : 1000

Enter the title of the book : ansi c

Enter the name of the author : ronak

Enter the release year of the book : 2003

--------

Title : dsa

Author : divya

Release year : 1999

Price : 900.000000

-----------

--------

Title : ansi c

Author : ronak

Release year : 2003

Price : 1000.000000

-----------

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);

}