1

#include <stdio.h>

int main() {

int a, b, sum;

printf("Enter two integers: ");

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

sum = a + b;

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

return 0;

}

2

#include <stdio.h>

int main() {

int x, y, temp;

printf("Enter two numbers: ");

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

temp = x;

x = y;

y = temp;

printf("Swapped values: x = %d, y = %d\n", x, y);

return 0;

}

3

#include <stdio.h>

int main() {

int x, y;

printf("Enter two numbers: ");

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

x = x + y;

y = x - y;

x = x - y;

printf("Swapped values: x = %d, y = %d\n", x, y);

return 0;

}

4

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

printf("ASCII value: %d\n", ch);

return 0;

}

5

#include <stdio.h>

int main() {

float length, width, area, perimeter;

printf("Enter length and width: ");

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

area = length \* width;

perimeter = 2 \* (length + width);

printf("Area: %.2f, Perimeter: %.2f\n", area, perimeter);

return 0;

}

6

#include <stdio.h>

int main() {

float principal, rate, time, interest;

printf("Enter principal, rate, and time: ");

scanf("%f %f %f", &principal, &rate, &time);

interest = (principal \* rate \* time) / 100;

printf("Simple Interest: %.2f\n", interest);

return 0;

}

7

#include <stdio.h>

int main() {

float celsius, fahrenheit;

printf("Enter temperature in Celsius: ");

scanf("%f", &celsius);

fahrenheit = (celsius \* 9/5) + 32;

printf("Temperature in Fahrenheit: %.2f\n", fahrenheit);

return 0;

}

8

#include <stdio.h>

int main() {

int dividend, divisor, quotient, remainder;

printf("Enter dividend and divisor: ");

scanf("%d %d", &dividend, &divisor);

quotient = dividend / divisor;

remainder = dividend % divisor;

printf("Quotient: %d, Remainder: %d\n", quotient, remainder);

return 0;

}

9

#include <stdio.h>

int main() {

int num;

printf("Enter an integer: ");

scanf("%d", &num);

if (num % 2 == 0)

printf("%d is even\n", num);

else

printf("%d is odd\n", num);

return 0;

}

10

#include <stdio.h>

int main() {

int num, square, cube;

printf("Enter a number: ");

scanf("%d", &num);

square = num \* num;

cube = num \* num \* num;

printf("Square: %d, Cube: %d\n", square, cube);

return 0;

}

# ASSIGMENT -2

#include <stdio.h>

1. Check if a number is positive, negative, or zero

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num > 0)

printf("Positive\n");

else if (num < 0)

printf("Negative\n");

else

printf("Zero\n");

return 0;

}

```

2. Find the largest among three numbers

#include <stdio.h>

int main() {

int a, b, c;

printf("Enter three numbers: ");

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

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

printf("Largest number: %d\n", a);

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

printf("Largest number: %d\n", b);

else

printf("Largest number: %d\n", c);

return 0;

}

3. Check if a year is a leap year

#include <stdio.h>

int main() {

int year;

printf("Enter a year: ");

scanf("%d", &year);

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))

printf("Leap year\n");

else

printf("Not a leap year\n");

return 0;

}

4. Check whether a character is a vowel or consonant

#include <stdio.h>

int main() {

char ch;

printf("Enter a character: ");

scanf(" %c", &ch);

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')

printf("Vowel\n");

else

printf("Consonant\n");

return 0;

}

5. Assign grades based on marks

#include <stdio.h>

int main() {

int marks;

printf("Enter marks: ");

scanf("%d", &marks);

if (marks >= 90)

printf("Grade A\n");

else if (marks >= 80)

printf("Grade B\n");

else if (marks >= 70)

printf("Grade C\n");

else if (marks >= 60)

printf("Grade D\n");

else

printf("Grade F\n");

return 0;

}

6. Check whether a number is divisible by 5 and 11

c

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num % 5 == 0 && num % 11 == 0)

printf("Divisible by 5 and 11\n");

else

printf("Not divisible by 5 and 11\n");

return 0;

}

```

7. Find the absolute value of a number

```c

#include <stdio.h>

#include <stdlib.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

printf("Absolute value: %d\n", abs(num));

return 0;

}

8. Menu-driven program for arithmetic operations

```c

#include <stdio.h>

int main() {

int choice, a, b;

printf("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n");

printf("Enter your choice: ");

scanf("%d", &choice);

printf("Enter two numbers: ");

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

switch (choice) {

case 1: printf("Result: %d\n", a + b); break;

case 2: printf("Result: %d\n", a - b); break;

case 3: printf("Result: %d\n", a \* b); break;

case 4: if (b != 0) printf("Result: %.2f\n", (float)a / b); else printf("Division by zero error\n"); break;

default: printf("Invalid choice\n");

}

return 0;

}

9. Find roots of a quadratic equation

#include <stdio.h>

#include <math.h>

int main() {

float a, b, c, discriminant, root1, root2;

printf("Enter coefficients a, b, and c: ");

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

discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0) {

root1 = (-b + sqrt(discriminant)) / (2 \* a);

root2 = (-b - sqrt(discriminant)) / (2 \* a);

printf("Roots are: %.2f and %.2f\n", root1, root2);

} else if (discriminant == 0) {

root1 = -b / (2 \* a);

printf("Root is: %.2f\n", root1);

} else {

printf("Roots are complex numbers.\n");

}

return 0;

}

10. Find the number of digits in a number

#include <stdio.h>

int main() {

int num, count = 0;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

num /= 10;

count++;

}

printf("Number of digits: %d\n", count);

return 0;

}

1. Print numbers from 1 to 100

#include <stdio.h>

int main() {

for (int i = 1; i <= 100; i++) {

printf("%d ", i);

}

return 0;

}

2. Print even numbers from 1 to 50

#include <stdio.h>

int main() {

for (int i = 2; i <= 50; i += 2) {

printf("%d ", i);

}

return 0;

}

3. Find the factorial of a number

#include <stdio.h>

int main() {

int num, fact = 1;

printf("Enter a number: ");

scanf("%d", &num);

for (int i = 1; i <= num; i++) {

fact \*= i;

}

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

return 0;

}

4. Calculate the sum of digits of a number

#include <stdio.h>

int main() {

int num, sum = 0, digit;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

digit = num % 10;

sum += digit;

num /= 10;

}

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

return 0;

}

5. Reverse a number

#include <stdio.h>

int main() {

int num, rev = 0, digit;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

digit = num % 10;

rev = rev \* 10 + digit;

num /= 10;

}

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

return 0;

}

6. Check whether a number is a palindrome

#include <stdio.h>

int main() {

int num, temp, rev = 0, digit;

printf("Enter a number: ");

scanf("%d", &num);

temp = num;

while (temp != 0) {

digit = temp % 10;

rev = rev \* 10 + digit;

temp /= 10;

}

if (num == rev)

printf("Palindrome\n");

else

printf("Not a palindrome\n");

return 0;

}

7. Print multiplication table of a number

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

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

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

}

return 0;

}

8. Count the number of digits in a number

#include <stdio.h>

int main() {

int num, count = 0;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

num /= 10;

count++;

}

printf("Number of digits: %d\n", count);

return 0;

}

9. Print the Fibonacci series up to n terms

#include <stdio.h>

int main() {

int n, first = 0, second = 1, next;

printf("Enter number of terms: ");

scanf("%d", &n);

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

printf("%d ", first);

next = first + second;

first = second;

second = next;

}

return 0;

}

10. Calculate the sum of the first n natural numbers

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

sum = (n \* (n + 1)) / 2;

printf("Sum of first %d natural numbers: %d\n", n, sum);

return 0;

}

# ASSIGNMENT 3

Here are the C programs for your requested tasks:

1.Read and print elements of an array

#include <stdio.h>

int main() {

int n;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

printf("Array elements: ");

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

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

return 0;

}

2. Find the sum of elements of an array

#include <stdio.h>

int main() {

int n, sum = 0;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

sum += arr[i];

}

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

return 0;

}

3. Find the maximum and minimum element in an array

#include <stdio.h>

int main() {

int n, max, min;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

max = min = arr[0];

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

if (arr[i] > max) max = arr[i];

if (arr[i] < min) min = arr[i];

}

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

return 0;

}

4. Reverse an array

#include <stdio.h>

int main() {

int n;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

printf("Reversed array: ");

for (int i = n - 1; i >= 0; i--)

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

return 0;

}

5. Search for an element in an array (Linear Search)

#include <stdio.h>

int main() {

int n, key, found = 0;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

printf("Enter element to search: ");

scanf("%d", &key);

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

if (arr[i] == key) {

found = 1;

printf("Element found at index %d\n", i);

break;

}

}

if (!found) printf("Element not found.\n");

return 0;

}

6. Sort an array in ascending order

#include <stdio.h>

int main() {

int n, temp;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

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

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

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

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("Sorted array: ");

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

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

return 0;

}

7. Insert an element in an array

#include <stdio.h>

int main() {

int n, pos, value;

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

scanf("%d", &n);

int arr[n+1];

printf("Enter elements: ");

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

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

printf("Enter position and value to insert: ");

scanf("%d %d", &pos, &value);

for (int i = n; i >= pos; i--)

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

arr[pos-1] = value;

printf("Updated array: ");

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

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

return 0;

}

8. Delete an element from an array

#include <stdio.h>

int main() {

int n, pos;

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

scanf("%d", &n);

int arr[n];

printf("Enter elements: ");

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

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

printf("Enter position of element to delete: ");

scanf("%d", &pos);

for (int i = pos - 1; i < n - 1; i++)

arr[i] = arr[i+1];

printf("Updated array: ");

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

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

return 0;

}

9. Find frequency of elements in an array

#include <stdio.h>

int main() {

int n, i, j, count;

printf("Enter number of elements: ");

scanf("%d", &n);

int arr[n], freq[n];

printf("Enter elements: ");

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

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

freq[i] = -1;

}

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

count = 1;

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

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

count++;

freq[j] = 0;

}

}

if (freq[i] != 0) freq[i] = count;

}

printf("Element frequencies:\n");

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

if (freq[i] != 0)

printf("%d appears %d times\n", arr[i], freq[i]);

}

return 0;

}

10. Merge two arrays

#include <stdio.h>

int main() {

int n1, n2;

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

scanf("%d", &n1);

int arr1[n1];

printf("Enter elements of first array: ");

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

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

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

scanf("%d", &n2);

int arr2[n2], merged[n1 + n2];

printf("Enter elements of second array: ");

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

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

for (int i = 0; i < n1; i++) merged[i] = arr1[i];

for (int i = 0; i < n2; i++) merged[n1 + i] = arr2[i];

printf("Merged array: ");

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

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

return 0;

}

# ASSIGNMENT 5

1. Find the Length of a String (Without `strlen()`)

#include <stdio.h>

int stringLength(char str[]) {

int count = 0;

while (str[count] != '\0') {

count++;

}

return count;

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

printf("Length of string: %d\n", stringLength(str));

return 0;

}

```

---

2. Copy One String to Another

#include <stdio.h>

void copyString(char source[], char destination[]) {

int i = 0;

while (source[i] != '\0') {

destination[i] = source[i];

i++;

}

destination[i] = '\0';

}

int main() {

char str1[100], str2[100];

printf("Enter a string: ");

gets(str1);

copyString(str1, str2);

printf("Copied string: %s\n", str2);

return 0;

}

3. Concatenate Two Strings

#include <stdio.h>

void concatenate(char str1[], char str2[]) {

int i = 0, j = 0;

while (str1[i] != '\0') {

i++;

}

while (str2[j] != '\0') {

str1[i] = str2[j];

i++;

j++;

}

str1[i] = '\0';

}

int main() {

char str1[100], str2[100];

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

concatenate(str1, str2);

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

return 0;

}

4. Compare Two Strings

#include <stdio.h>

int compareStrings(char str1[], char str2[]) {

int i = 0;

while (str1[i] != '\0' && str2[i] != '\0') {

if (str1[i] != str2[i]) {

return str1[i] - str2[i];

}

i++;

}

return str1[i] - str2[i];

}

int main() {

char str1[100], str2[100];

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

int result = compareStrings(str1, str2);

if (result == 0)

printf("Strings are equal\n");

else

printf("Strings are different\n");

return 0;

}

Here are the next set of programs for your string manipulation exercises:

---

5. Count Vowels and Consonants in a String

#include <stdio.h>

void countVowelsConsonants(char str[]) {

int vowels = 0, consonants = 0;

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

char ch = str[i];

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z')) {

if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U' ||

ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowels++;

} else {

consonants++;

}

}

}

printf("Vowels: %d\nConsonants: %d\n", vowels, consonants);

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

countVowelsConsonants(str);

return 0;

}

6. Convert Lowercase to Uppercase and Vice Versa

#include <stdio.h>

void convertCase(char str[]) {

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

if (str[i] >= 'a' && str[i] <= 'z') {

str[i] -= 32;

} else if (str[i] >= 'A' && str[i] <= 'Z') {

str[i] += 32;

}

}

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

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

convertCase(str);

return 0;

}

7. Check if a String is Palindrome

#include <stdio.h>

#include <string.h>

int isPalindrome(char str[]) {

int length = strlen(str);

for (int i = 0; i < length / 2; i++) {

if (str[i] != str[length - i - 1]) {

return 0;

}

}

return 1;

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

if (isPalindrome(str))

printf("The string is a palindrome.\n");

else

printf("The string is not a palindrome.\n");

return 0;

}

8. Reverse a String

#include <stdio.h>

#include <string.h>

void reverseString(char str[]) {

int length = strlen(str);

for (int i = 0; i < length / 2; i++) {

char temp = str[i];

str[i] = str[length - i - 1];

str[length - i - 1] = temp;

}

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

reverseString(str);

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

return 0;

}

```

9. Count Words in a String

#include <stdio.h>

int countWords(char str[]) {

int count = 0, i = 0;

while (str[i] != '\0') {

if ((str[i] == ' ' || str[i] == '\n' || str[i] == '\t') && str[i + 1] != ' ' && str[i + 1] != '\0') {

count++;

}

i++;

}

return count + 1; // Adding 1 for the first word

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

printf("Word count: %d\n", countWords(str));

return 0;

}

10. Find the Frequency of Each Character in a String

#include <stdio.h>

void characterFrequency(char str[]) {

int freq[256] = {0}; // ASCII size

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

freq[(int)str[i]]++;

}

printf("Character frequencies:\n");

for (int i = 0; i < 256; i++) {

if (freq[i] > 0) {

printf("%c: %d\n", i, freq[i]);

}

}

}

int main() {

char str[100];

printf("Enter a string: ");

gets(str);

characterFrequency(str);

return 0;

}

# ASSIGNMENT -6

Here’s the program with clear headings for each function, Uday! 🚀

---

## 1. Function to Find Factorial of a Number

```c

#include <stdio.h>

long long factorial(int n) {

return (n == 0 || n == 1) ? 1 : n \* factorial(n - 1);

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

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

return 0;

}

```

---

## 2. Function to Check Whether a Number is Prime

```c

#include <stdio.h>

int isPrime(int num) {

if (num < 2) return 0;

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

if (num % i == 0) return 0;

return 1;

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

printf("%d is %s\n", num, isPrime(num) ? "a prime number" : "not a prime number");

return 0;

}

3. Function to Calculate Power Using Recursion

#include <stdio.h>

long long power(int base, int exp) {

return (exp == 0) ? 1 : base \* power(base, exp - 1);

}

int main() {

int base, exp;

printf("Enter base and exponent: ");

scanf("%d %d", &base, &exp);

printf("%d^%d = %lld\n", base, exp, power(base, exp));

return 0;

}

4. Function to Check Palindrome Number Using Recursion

#include <stdio.h>

#include <string.h>

int isPalindrome(char str[], int start, int end) {

return (start >= end) ? 1 : (str[start] == str[end] && isPalindrome(str, start + 1, end - 1));

}

int main() {

char str[100];

printf("Enter a number as string: ");

scanf("%s", str);

printf("%s is %s\n", str, isPalindrome(str, 0, strlen(str) - 1) ? "a palindrome" : "not a palindrome");

return 0;

}

5. Function to Calculate nCr (Combinations)

#include <stdio.h>

long long factorial(int n) {

return (n == 0 || n == 1) ? 1 : n \* factorial(n - 1);

}

long long nCr(int n, int r) {

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

}

int main() {

int n, r;

printf("Enter n and r: ");

scanf("%d %d", &n, &r);

printf("nCr(%d, %d) = %lld\n", n, r, nCr(n, r));

return 0;

}

```

---

## 6. Program to Demonstrate Call by Value and Call by Reference

```c

#include <stdio.h>

void callByValue(int a) {

a += 10;

printf("Inside callByValue: %d\n", a);

}

void callByReference(int \*a) {

\*a += 10;

printf("Inside callByReference: %d\n", \*a);

}

int main() {

int num = 20;

printf("Original value: %d\n", num);

callByValue(num);

printf("After callByValue: %d\n", num);

callByReference(&num);

printf("After callByReference: %d\n", num);

return 0;

}

```

---

## 7. Function to Swap Two Numbers

```c

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

printf("Enter two numbers: ");

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

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

swapNumbers(&x, &y);

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

return 0;

}

8. Recursive Function to Find the nth Fibonacci Number

#include <stdio.h>

int fibonacci(int n) {

return (n == 0) ? 0 : (n == 1) ? 1 : fibonacci(n - 1) + fibonacci(n - 2);

}

int main() {

int n;

printf("Enter position for Fibonacci: ");

scanf("%d", &n);

printf("Fibonacci number at position %d: %d\n", n, fibonacci(n));

return 0;

}

9. Program to Find GCD and LCM Using Functions

#include <stdio.h>

int gcd(int a, int b) {

return (b == 0) ? a : gcd(b, a % b);

}

int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

int main() {

int num1, num2;

printf("Enter two numbers: ");

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

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

printf("LCM of %d and %d: %d\n", num1, num2, lcm(num1, num2));

return 0;

}

10. Program to Demonstrate Global and Local Variables

#include <stdio.h>

int globalVar = 10; // Global Variable

void function() {

int localVar = 5; // Local Variable

printf("Inside function: Local Variable = %d, Global Variable = %d\n", localVar, globalVar);

}

int main() {

function();

printf("Inside main: Global Variable = %d\n", globalVar);

return 0;

}

# ASSIGNMENT -7

1. Access Array Elements Using Pointers

#include <stdio.h>

int main() {

int arr[] = {10, 20, 30, 40, 50};

int \*ptr = arr;

printf("Array elements using pointers:\n");

for (int i = 0; i < 5; i++) {

printf("%d ", \*(ptr + i));

}

return 0;

}

2. Swap Two Numbers Using Pointers

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

printf("Enter two numbers: ");

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

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

swap(&x, &y);

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

return 0;

}

3. Add Two Numbers Using Pointers

```c

#include <stdio.h>

int main() {

int a, b, sum;

int \*ptr1 = &a, \*ptr2 = &b;

printf("Enter two numbers: ");

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

sum = \*ptr1 + \*ptr2;

printf("Sum = %d\n", sum);

return 0;

}

4. Find the Length of a String Using Pointers

#include <stdio.h>

int stringLength(char \*str) {

int count = 0;

while (\*str) {

count++;

str++;

}

return count;

}

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

printf("Length of string: %d\n", stringLength(str));

return 0;

}

5. Reverse a String Using Pointers

#include <stdio.h>

#include <string.h>

void reverseString(char \*str) {

char \*start = str;

char \*end = str + strlen(str) - 1;

while (start < end) {

char temp = \*start;

\*start = \*end;

\*end = temp;

start++;

end--;

}

}

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

reverseString(str);

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

return 0;

}

6. Count Vowels Using Pointers

#include <stdio.h>

int countVowels(char \*str) {

int count = 0;

while (\*str) {

if (\*str == 'a' || \*str == 'e' || \*str == 'i' || \*str == 'o' || \*str == 'u' ||

\*str == 'A' || \*str == 'E' || \*str == 'I' || \*str == 'O' || \*str == 'U') {

count++;

}

str++;

}

return count;

}

int main() {

char str[100];

printf("Enter a string: ");

scanf("%s", str);

printf("Vowel count: %d\n", countVowels(str));

return 0;

}

7. Demonstrate Pointer to Pointer

#include <stdio.h>

int main() {

int num = 10;

int \*ptr = &num;

int \*\*ptr2 = &ptr;

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

printf("Value using pointer: %d\n", \*ptr);

printf("Value using pointer to pointer: %d\n", \*\*ptr2);

return 0;

}

```

---

8. Allocate Memory Using `malloc()` and Free It

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*ptr, n;

printf("Enter number of elements: ");

scanf("%d", &n);

ptr = (int\*) malloc(n \* sizeof(int));

if (ptr == NULL) {

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

return 1;

}

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

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

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

printf("Stored elements: ");

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

printf("%d ", \*(ptr + i));

free(ptr);

return 0;

}

9. Sort an Array Using Pointer Notation

```c

#include <stdio.h>

void sortArray(int \*arr, int n) {

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

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

if (\*(arr + i) > \*(arr + j)) {

int temp = \*(arr + i);

\*(arr + i) = \*(arr + j);

\*(arr + j) = temp;

}

}

}

}

int main() {

int arr[100], n;

printf("Enter number of elements: ");

scanf("%d", &n);

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

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

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

sortArray(arr, n);

printf("Sorted array: ");

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

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

return 0;

}

# ASSIGNMENT 8

Here are all 10 programs in a single block, Uday! These exercises will solidify your understanding of \*\*structures and unions\*\* in C. 🚀

---

1. Define a Structure for Student Record and Print Details

#include <stdio.h>

struct Student {

char name[50];

int roll\_no;

float marks;

};

int main() {

struct Student s1 = {"Uday", 101, 89.5};

printf("Student Details:\n");

printf("Name: %s\nRoll No: %d\nMarks: %.2f\n", s1.name, s1.roll\_no, s1.marks);

return 0;

}

2. Store and Display Employee Details Using Structures

#include <stdio.h>

struct Employee {

char name[50];

int id;

float salary;

};

int main() {

struct Employee e1 = {"Raj", 1001, 55000.75};

printf("Employee Details:\n");

printf("Name: %s\nID: %d\nSalary: %.2f\n", e1.name, e1.id, e1.salary);

return 0;

}

3. Pass a Structure to a Function

#include <stdio.h>

struct Student {

char name[50];

int roll\_no;

float marks;

};

void display(struct Student s) {

printf("Name: %s\nRoll No: %d\nMarks: %.2f\n", s.name, s.roll\_no, s.marks);

}

int main() {

struct Student s1 = {"Amit", 102, 92.3};

display(s1);

return 0;

}

4. Store Multiple Student Records Using Array of Structures

#include <stdio.h>

struct Student {

char name[50];

int roll\_no;

float marks;

};

int main() {

struct Student students[3] = {

{"Amit", 101, 88.5},

{"Neha", 102, 91.2},

{"Raj", 103, 85.0}

};

printf("Student Records:\n");

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

printf("Name: %s | Roll No: %d | Marks: %.2f\n", students[i].name, students[i].roll\_no, students[i].marks);

}

return 0;

}

5. Demonstrate Nested Structures

#include <stdio.h>

struct Address {

char city[50];

int pin;

};

struct Employee {

char name[50];

int id;

float salary;

struct Address addr;

};

int main() {

struct Employee emp = {"Raj", 1002, 60000.25, {"Mumbai", 400001}};

printf("Employee Details:\nName: %s\nID: %d\nSalary: %.2f\nCity: %s\nPIN: %d\n",

emp.name, emp.id, emp.salary, emp.addr.city, emp.addr.pin);

return 0;

}

6. Calculate Total and Average Marks Using Structures

#include <stdio.h>

struct Student {

char name[50];

float marks[3];

};

int main() {

struct Student s = {"Amit", {85.5, 90.0, 78.0}};

float total = 0;

for (int i = 0; i < 3; i++) total += s.marks[i];

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

printf("Average Marks: %.2f\n", total / 3);

return 0;

}

7. Find the Highest Marks Among Students

#include <stdio.h>

struct Student {

char name[50];

float marks;

};

int main() {

struct Student students[3] = {

{"Amit", 88.5},

{"Neha", 91.2},

{"Raj", 85.0}

};

struct Student top = students[0];

for (int i = 1; i < 3; i++) {

if (students[i].marks > top.marks)

top = students[i];

}

printf("Highest Scorer: %s with %.2f marks\n", top.name, top.marks);

return 0;

}

8. Sort Student Records by Name Using Structures

#include <stdio.h>

#include <string.h>

struct Student {

char name[50];

float marks;

};

void sortStudents(struct Student students[], int n) {

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

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

if (strcmp(students[i].name, students[j].name) > 0) {

struct Student temp = students[i];

students[i] = students[j];

students[j] = temp;

}

}

}

}

int main() {

struct Student students[3] = {

{"Amit", 88.5},

{"Neha", 91.2},

{"Raj", 85.0}

};

sortStudents(students, 3);

printf("Sorted Student Records:\n");

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

printf("Name: %s | Marks: %.2f\n", students[i].name, students[i].marks);

}

return 0;

}

9. Using Union to Store Data of Different Types

#include <stdio.h>

union Data {

int i;

float f;

char str[20];

};

int main() {

union Data data;

data.i = 10;

printf("Integer: %d\n", data.i);

data.f = 22.5;

printf("Float: %.2f\n", data.f);

strcpy(data.str, "Hello");

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

return 0;

}

10. Compare Structure vs Union with a Sample Program

#include <stdio.h>

#include <string.h>

struct StructureExample {

int i;

float f;

char str[20];

};

union UnionExample {

int i;

float f;

char str[20];

};

int main() {

struct StructureExample s = {10, 22.5, "Hello"};

union UnionExample u;

u.i = 10;

printf("Union Integer: %d\n", u.i);

u.f = 22.5;

printf("Union Float: %.2f\n", u.f);

strcpy(u.str, "Hello");

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

printf("\nStructure stores all values, whereas Union overwrites.\n");

return 0;

}

# ASSIGMENT -9

Great list, Uday! These file-handling tasks will definitely strengthen your understanding of working with files in C. Here are solutions for each:

### 1. Create and Write to a File

```c

#include <stdio.h>

int main() {

FILE \*fp = fopen("example.txt", "w"); // Open file in write mode

if (fp == NULL) {

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

return 1;

}

fprintf(fp, "Hello, this is a sample text file.\n");

fclose(fp);

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

return 0;

}

```

### 2. Read Contents of a File

```c

#include <stdio.h>

int main() {

FILE \*fp = fopen("example.txt", "r");

if (fp == NULL) {

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

return 1;

}

char ch;

while ((ch = fgetc(fp)) != EOF) {

putchar(ch);

}

fclose(fp);

return 0;

}

### 3. Count Number of Lines in a Fil

#include <stdio.h>

int main() {

FILE \*fp = fopen("example.txt", "r");

if (fp == NULL) {

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

return 1;

}

char ch;

int lines = 0;

while ((ch = fgetc(fp)) != EOF) {

if (ch == '\n') {

lines++;

}

}

fclose(fp);

printf("Total lines: %d\n", lines);

return 0;

}

4. Copy Contents from One File to Another

#include <stdio.h>

int main() {

FILE \*src = fopen("source.txt", "r");

FILE \*dest = fopen("destination.txt", "w");

if (src == NULL || dest == NULL) {

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

return 1;

}

char ch;

while ((ch = fgetc(src)) != EOF) {

fputc(ch, dest);

}

fclose(src);

fclose(dest);

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

return 0;

}

### 5. Append Text to a File

c

#include <stdio.h>

int main() {

FILE \*fp = fopen("example.txt", "a");

if (fp == NULL) {

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

return 1;

}

fprintf(fp, "Appended line.\n");

fclose(fp);

printf("Text appended successfully.\n");

return 0;

}

6. Count Vowels in a File

#include <stdio.h>

int main() {

FILE \*fp = fopen("example.txt", "r");

if (fp == NULL) {

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

return 1;

}

char ch;

int vowels = 0;

while ((ch = fgetc(fp)) != EOF) {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

vowels++;

}

}

fclose(fp);

printf("Total vowels: %d\n", vowels);

return 0;

}

7. Read Integers from a File and Find the Sum

#include <stdio.h>

int main() {

FILE \*fp = fopen("numbers.txt", "r");

if (fp == NULL) {

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

return 1;

}

int num, sum = 0;

while (fscanf(fp, "%d", &num) == 1) {

sum += num;

}

fclose(fp);

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

return 0;

}

8. Read a Structure from a File

#include <stdio.h>

struct Student {

char name[50];

int age;

};

int main() {

FILE \*fp = fopen("student.dat", "rb");

if (fp == NULL) {

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

return 1;

}

struct Student s;

fread(&s, sizeof(struct Student), 1, fp);

fclose(fp);

printf("Name: %s\nAge: %d\n", s.name, s.age);

return 0;

}

### 9. Sort Names Stored in a File

#include <stdio.h>

#include <string.h>

int main() {

FILE \*fp = fopen("names.txt", "r");

if (fp == NULL) {

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

return 1;

}

char names[100][50];

int count = 0;

while (fscanf(fp, "%s", names[count]) == 1) {

count++;

}

fclose(fp);

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

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

if (strcmp(names[i], names[j]) > 0) {

char temp[50];

strcpy(temp, names[i]);

strcpy(names[i], names[j]);

strcpy(names[j], temp);

}

}

}

fp = fopen("sorted\_names.txt", "w");

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

fprintf(fp, "%s\n", names[i]);

}

fclose(fp);

printf("Names sorted successfully.\n");

return 0;

}

10. Search for a Word in a File

#include <stdio.h>

#include <string.h>

int main() {

FILE \*fp = fopen("example.txt", "r");

if (fp == NULL) {

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

return 1;

}

char word[50], temp[1000];

printf("Enter word to search: ");

scanf("%s", word);

int found = 0;

while (fscanf(fp, "%s", temp) == 1) {

if (strcmp(temp, word) == 0) {

found = 1;

break;

}

}

fclose(fp);

if (found) {

printf("Word found in file.\n");

} else {

printf("Word not found in file.\n");

}

return 0;

}