BIT WISE OPS

#include <stdio.h>

int main() {

    int choice, num1, num2, result, shift;

    do {

        printf("\nMenu:\n");

        printf("1. Bitwise AND\n");

        printf("2. Bitwise OR\n");

        printf("3. Right Shift\n");

        printf("4. Left Shift\n");

        printf("5. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                printf("Enter two integers: ");

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

                result = num1 & num2;

                printf("Result of Bitwise AND: %d\n", result);

                break;

            case 2:

                printf("Enter two integers: ");

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

                result = num1 | num2;

                printf("Result of Bitwise OR: %d\n", result);

                break;

            case 3:

                printf("Enter an integer and shift value: ");

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

                result = num1 >> shift;

                printf("Result of Right Shift: %d\n", result);

                break;

            case 4:

                printf("Enter an integer and shift value: ");

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

                result = num1 << shift;

                printf("Result of Left Shift: %d\n", result);

                break;

            case 5:                 printf("Exiting program.\n");

                break;

            default:

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

    } while (choice != 5);

    return 0; }

array ops

#include <stdio.h>

int main()

{

    int choice, n, i, sum, min, element, found;

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

    scanf("%d", &n);

    int arr[n];

    // Read the array

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

    for (i = 0; i < n; i++)

    {

        printf("Enter element %d: ", i + 1);

        scanf("%d", &arr[i]);

    }

    do

    {

        printf("\nMenu:\n");

        printf("1. Print Addition of Array Elements\n");

        printf("2. Find Minimum Number from Array\n");

        printf("3. Search an Element from Array\n");

        printf("4. Print the Input Array\n");

        printf("5. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            sum = 0;

            for (i = 0; i < n; i++)

            {

                sum += arr[i];

            }

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

            break;

        case 2:

            min = arr[0];

            for (i = 1; i < n; i++)

            {

                if (arr[i] < min)

                {

                    min = arr[i];

                }

            }

            printf("The minimum number in the array is: %d\n", min);

            break;

        case 3:

            printf("Enter the element to search: ");

            scanf("%d", &element);

            found = 0;

            for (i = 0; i < n; i++)

            {

                if (arr[i] == element)

                {

                    printf("Element %d is at position %d (index is %d).\n", element, i + 1, i);

                    found = 1;

                    break;

                }

            }

            if (!found)

            {

                printf("Element %d not found in the array.\n", element);

            }

            break;

        case 4:

            printf("The array elements are: ");

            for (i = 0; i < n; i++)

            {

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

            }

            printf("\n");

            break;

        case 5:

            printf("Exiting the program.\n");

            break;

        default:

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

        }

    } while (choice != 5);

    return 0;

}

String ops

// Menu-driven program to perform operations on strings.

#include <stdio.h>

void printStringLength(char str[]) {

    int length = 0;

    while (str[length] != '\0') {

        length++;

    }

    printf("Length of the string: %d\n", length);

}

void reverseString(char str[]) {

    int length = 0, i;

    char reversed[100];

    while (str[length] != '\0') {

        length++;

    }

    for (i = 0; i < length; i++) {

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

    }

    reversed[length] = '\0';

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

}

void copyString(char str[]) {

    char copy[100];

    int i = 0;

    while (str[i] != '\0') {

        copy[i] = str[i];

        i++;

    }

    copy[i] = '\0';

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

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

}

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

    int i = 0;

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

        if (str1[i] != str2[i]) {

            if (str1[i] < str2[i]) {

                printf("\"%s\" is smaller than \"%s\"\n", str1, str2);

                return;

            } else {

                printf("\"%s\" is greater than \"%s\"\n", str1, str2);

                return;

            }

        }

        i++;

    }

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

        printf("\"%s\" and \"%s\" are equal\n", str1, str2);

    } else if (str1[i] == '\0') {

        printf("\"%s\" is smaller than \"%s\"\n", str1, str2);

    } else {

        printf("\"%s\" is greater than \"%s\"\n", str1, str2);

    }

}

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

    char concatenated[200];

    int i = 0, j = 0;

    while (str1[i] != '\0') {

        concatenated[i] = str1[i];

        i++;

    }

    while (str2[j] != '\0') {

        concatenated[i] = str2[j];

        i++;

        j++;

    }

    concatenated[i] = '\0';

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

}

int main() {

    int choice;

    char str1[100], str2[100];

    do {

        printf("\nMenu:\n");

        printf("1. Print String Length\n");

        printf("2. Print String in Reverse Order\n");

        printf("3. Copy Given String\n");

        printf("4. Compare Two Strings\n");

        printf("5. Concatenate Two Strings\n");

        printf("6. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

switch (choice) {

            case 1:

                printf("Enter a string: ");

                scanf("%s", str1);

                printStringLength(str1);

                break;

            case 2:

                printf("Enter a string: ");

                scanf("%s", str1);

                reverseString(str1);

                break;

            case 3:

                printf("Enter a string: ");

                scanf("%s", str1);

                copyString(str1);

                break;

            case 4:

                printf("Enter first string: ");

                scanf("%s", str1);

                printf("Enter second string: ");

                scanf("%s", str2);

                compareStrings(str1, str2);

                break;

            case 5:

                printf("Enter first string: ");

                scanf("%s", str1);

                printf("Enter second string: ");

                scanf("%s", str2);

                concatenateStrings(str1, str2);

                break;

            case 6:

                printf("Exiting program.\n");

                break;

            default:

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

        }

    } while (choice != 6);

    return 0;

}

Power recursion, decimal binary, number prime, display prime

#include <stdio.h>

// Function to calculate power using recursion

int power(int base, int exp) {

    if (exp == 0)

        return 1;

    return base \* power(base, exp - 1);

}

// Function to check if a number is prime

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;

}

// Function to display prime numbers in an interval

void displayPrimes(int start, int end) {

    printf("Prime numbers between %d and %d are: ", start, end);

    for (int i = start; i <= end; i++) {

        if (isPrime(i))

            printf("%d ", i);

    }

    printf("\n");

}

// Function to convert decimal to binary

void decimalToBinary(int num) {

    if (num > 1)

        decimalToBinary(num / 2);

    printf("%d", num % 2);

}

int main() {

    int choice, base, exp, start, end, num;

    do {

        printf("\nMenu:\n");

        printf("1. Calculate power using recursion\n");

        printf("2. Display all prime numbers between two intervals\n");

        printf("3. Convert decimal to binary\n");

        printf("4. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                printf("Enter base and exponent: ");

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

                printf("Result: %d\n", power(base, exp));

                break;

            case 2:

                printf("Enter two numbers (start and end): ");

                scanf("%d %d", &start, &end);

                displayPrimes(start, end);

                break;

            case 3:

                printf("Enter a decimal number: ");

                scanf("%d", &num);

                printf("Binary representation: ");

                decimalToBinary(num);

                printf("\n");

                break;

            case 4:

                printf("Exiting program.\n");

                break;

            default:

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

        }

    } while (choice != 4);

    return 0;

}

String length, copy strings

#include <stdio.h>

// Function to find the length of a string using pointers

int strlength(char \*str) {

    int len = 0;

    while (\*str != '\0') {

        len++;

        str++;

    }

    return len;

}

// Function to copy one string into another using pointers

void copyStr(char \*source, char \*destination) {

    printf("\nCopying string...\n");

    while (\*source != '\0') {

        \*destination = \*source;

        source++;

        destination++;

    }

    \*destination = '\0'; // Null-terminate copied string

}

int main() {

    int choice;

    char str1[100], str2[100];

    do {

        printf("\n==== String Operations Menu ====\n");

        printf("1. Find length of a string\n");

        printf("2. Copy one string into another\n");

        printf("3. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                printf("\nEnter a string: ");

                fgets(str1, sizeof(str1), stdin);

                // Remove newline character if present

                int len = strlength(str1);

                if (str1[len - 1] == '\n') {

                    str1[len - 1] = '\0';

                    len--;  // Update length

                }

                printf("Length of the string: %d\n", len);

                break;

            case 2:

                printf("\nEnter a string to copy: ");

                fgets(str1, sizeof(str1), stdin);

                // Remove newline character if present

                if (str1[strlength(str1) - 1] == '\n') {

                    str1[strlength(str1) - 1] = '\0';

                }

                copyStr(str1, str2);

                printf("Original String: %s\n", str1);

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

                break;

            case 3:

                printf("\nExiting program... Have a great day!\n");

                break;

            default:

                printf("\nInvalid choice! Please enter a number between 1 and 3.\n");

        }

    } while (choice != 3);

    return 0;

}

Swap numbers, array elemts pointers and dynamic memory alloc

#include <stdio.h>

// Function to find the length of a string using pointers

int strlength(char \*str) {

    int len = 0;

    while (\*str != '\0') {

        len++;

        str++;

    }

    return len;

}

// Function to copy one string into another using pointers

void copyStr(char \*source, char \*destination) {

    printf("\nCopying string...\n");

    while (\*source != '\0') {

        \*destination = \*source;

        source++;

        destination++;

    }

    \*destination = '\0'; // Null-terminate copied string

}

int main() {

    int choice;

    char str1[100], str2[100];

    do {

        printf("\n==== String Operations Menu ====\n");

        printf("1. Find length of a string\n");

        printf("2. Copy one string into another\n");

        printf("3. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice) {

            case 1:

                printf("\nEnter a string: ");

                fgets(str1, sizeof(str1), stdin);

                // Remove newline character if present

                int len = strlength(str1);

                if (str1[len - 1] == '\n') {

                    str1[len - 1] = '\0';

                    len--;  // Update length

                }

                printf("Length of the string: %d\n", len);

                break;

            case 2:

                printf("\nEnter a string to copy: ");

                fgets(str1, sizeof(str1), stdin);

                // Remove newline character if present

                if (str1[strlength(str1) - 1] == '\n') {

                    str1[strlength(str1) - 1] = '\0';

                }

                copyStr(str1, str2);

                printf("Original String: %s\n", str1);

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

                break;

            case 3:

                printf("\nExiting program... Have a great day!\n");

                break;

            default:

                printf("\nInvalid choice! Please enter a number between 1 and 3.\n");

        }

    } while (choice != 3);

    return 0;

}