A. BÀI TẬP MINH HỌA

Bài 1. Cho chuỗi s (|s| ≤ 1000). Hãy viết chương trình kiểm tra chuỗi có chứa ký tự số không?

Moi ban nhap chuoi s: abc12def

Chuoi “abc12def” co chua ky tu so.



#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 1000 // so luong ky tu toi da

/\*

Kiem tra chuoi co chua ky tu so

+ Vao: chuoi s

+ Ra : co = 0 (khong co) / 1 (co

+ Vi du: \*s = "abc12def" => co = 1

\*/

int CoKyTuSo(char \*s)

{

    int co;

    int len;

    len = strlen(s);                   // lay so ky tu chuoi s

    co = 0;                            // gia su chuoi s khong co chua ky tu

    for (int i = 0; i <= len - 1; i++) // duyet tu vi tri dau den cuoi

    {

        if (s[i] >= '0' && s[i] <= '9') // ky tu s[i] la ky tu so

        {

            co = 1; // bao hieu tim thay

            break;  // thoat lap

        } // if

    } // for

    return co;

} // CoKyTuSo

int main()

{

    /\* Khai bao bien \*/

    char sChuoi[MAX + 1] = {0}; // khoi tao chuoi rong

    int kq;

    /\* Nhap chuoi \*/

    printf("Moi ban nhap chuoi s: ");

    fflush(stdin); // xoa cache ban phim

    gets(sChuoi);

    /\* Xu ly\*/

    kq = CoKyTuSo(sChuoi);

    /\* Xuat du lieu \*/

    printf("Chuoi \"%s\" co chua ky tu so.", sChuoi);

    /\* Dung man hinh xem ket qua \*/

    system("PAUSE");

    return 0;

} // main

Bài 2. Một chuỗi được gọi là đối xứng nếu chúng ta đọc từ trái sang phải và từ phải sang trái đều giống

nhau. Ví dụ: “ABC” là không đối xứng, “ABA” là đối xứng.

Yêu cầu: Cho chuỗi s (|s| ≤ 1000). Hãy viết chương trình kiểm tra chuỗi s có đối xứng hay không?

Moi ban nhap chuoi s: abcba

Chuoi “abcba” la doi xung.



#include <stdio.h>

#include <string.h>

void KiemTraDoiXung(char \*s)

{

    char \*pDau = s;

    char \*pCuoi = s + strlen(s) - 1;

    int doixung = 1; // Giả sử chuỗi đối xứng

    while (pDau < pCuoi)

    {

        if (\*pDau != \*pCuoi)

        {

            doixung = 0; // Không đối xứng

            break;

        }

        pDau++;

        pCuoi--;

    }

    if (doixung)

    {

        printf("Chuoi \"%s\" la doi xung.\n", s);

    }

    else

    {

        printf("Chuoi \"%s\" khong doi xung.\n", s);

    }

}

int main()

{

    char s[1001]; // Định nghĩa mảng có thể chứa chuỗi tối đa 1000 ký tự

    printf("Moi ban nhap chuoi s: ");

    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = '\0'; // Loại bỏ ký tự xuống dòng nếu có

    KiemTraDoiXung(s);

    return 0;

}

Bài 3. Một từ là một ký tự liên tiếp không có khoảng trắng. Ví dụ: “ Bai tap “ là một chuỗi có hai từ

“Bai”, “tap”.

Yêu cầu: Cho chuỗi s (|s|≤1000). Hãy đếm số từ trong chuỗi.

Moi ban nhap chuoi s: Bai tap

Chuoi “ Bai tap ” co hai tu: “Bai”, “tap.



#include <stdio.h>

#include <string.h>

#include <ctype.h>

void DemSoTu(char \*s)

{

    int len = strlen(s);

    int i, dau = -1, cuoi = -1, soTu = 0;

    printf("Cac tu trong chuoi: ");

    for (i = 0; i < len; i++)

    {

        // Nếu là ký tự đầu của một từ

        if (!isspace(s[i]) && (i == 0 || isspace(s[i - 1])))

        {

            dau = i;

        }

        // Nếu là ký tự cuối của một từ

        if (!isspace(s[i]) && (i == len - 1 || isspace(s[i + 1])))

        {

            cuoi = i;

            soTu++;

            // In từ vừa tìm được

            printf("\"");

            for (int j = dau; j <= cuoi; j++)

            {

                printf("%c", s[j]);

            }

            printf("\", ");

        }

    }

    printf("\nChuoi \"%s\" co %d tu.\n", s, soTu);

}

int main()

{

    char s[1001]; // Định nghĩa mảng có thể chứa chuỗi tối đa 1000 ký tự

    printf("Moi ban nhap chuoi s: ");

    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = '\0'; // Loại bỏ ký tự xuống dòng nếu có

    DemSoTu(s);

    return 0;

}

B. BÀI TẬP PHẢI LÀM

Bài 1.



#include <stdio.h>

#include <string.h>

// Hàm hoán đổi 2 ký tự

void swap(char \*a, char \*b)

{

    char temp = \*a;

    \*a = \*b;

    \*b = temp;

}

// Hàm sắp xếp chuỗi tăng dần

void SapXepChuoi(char \*s)

{

    int len = strlen(s);

    for (int i = 0; i < len - 1; i++)

    {

        for (int j = i + 1; j < len; j++)

        {

            if (s[i] > s[j])

            {

                swap(&s[i], &s[j]);

            }

        }

    }

}

int main()

{

    char s[1001]; // Giới hạn chuỗi tối đa 1000 ký tự

    printf("Moi ban nhap chuoi s: ");

    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = '\0'; // Loại bỏ ký tự xuống dòng nếu có

    SapXepChuoi(s);

    printf("Chuoi sau khi sap xep: %s\n", s);

    return 0;

}

Bài 2.



#include <stdio.h>

#include <string.h>

// Hàm xóa ký tự tại vị trí k

void XoaKyTuTaiViTri(char \*s, int k)

{

    int len = strlen(s);

    // Kiểm tra vị trí hợp lệ

    if (k < 0 || k >= len)

    {

        printf("Vi tri khong hop le!\n");

        return;

    }

    // Dịch các ký tự sang trái từ vị trí k

    for (int i = k; i < len - 1; i++)

    {

        s[i] = s[i + 1];

    }

    s[len - 1] = '\0'; // Đặt ký tự kết thúc chuỗi

}

int main()

{

    char s[1001]; // Giới hạn chuỗi tối đa 1000 ký tự

    int k;

    // Nhập chuỗi

    printf("Moi ban nhap chuoi s: ");

    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = '\0'; // Loại bỏ ký tự xuống dòng nếu có

    // Nhập vị trí cần xóa

    printf("Moi ban nhap vi tri can xoa: ");

    scanf("%d", &k);

    XoaKyTuTaiViTri(s, k);

    printf("Chuoi sau khi xoa ky tu tai vi tri %d: %s\n", k, s);

    return 0;

}

Bài 3.



#include <stdio.h>

#include <string.h>

// Hàm chèn ký tự vào vị trí k

void ChenKyTuTaiViTri(char \*s, int k, char c)

{

    int len = strlen(s);

    // Kiểm tra vị trí hợp lệ

    if (k < 0 || k > len)

    {

        printf("Vi tri khong hop le!\n");

        return;

    }

    // Dịch các ký tự sang phải để tạo khoảng trống

    for (int i = len; i >= k; i--)

    {

        s[i + 1] = s[i];

    }

    // Chèn ký tự c vào vị trí k

    s[k] = c;

}

// Hàm main

int main()

{

    char s[1002]; // Giới hạn chuỗi tối đa 1001 ký tự (1000 + 1 cho '\0')

    int k;

    char c;

    // Nhập chuỗi

    printf("Moi ban nhap chuoi s: ");

    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = '\0'; // Loại bỏ ký tự xuống dòng nếu có

    // Nhập vị trí cần chèn

    printf("Moi ban nhap vi tri can chen: ");

    scanf("%d", &k);

    // Nhập ký tự cần chèn

    printf("Moi ban nhap ky tu chen: ");

    scanf(" %c", &c); // Khoảng trắng trước %c để bỏ qua ký tự xuống dòng còn lại trong bộ đệm

    // Gọi hàm chèn ký tự

    ChenKyTuTaiViTri(s, k, c);

    // Xuất chuỗi sau khi chèn

    printf("Chuoi sau khi them ky tu \"%c\" vao vi tri %d: %s\n", c, k, s);

    return 0;

}