BCS009 Assignment 2 石雨宸 D23090120503

Q1:

设顺序表 L 有 10 个整数。设计一个算法，以第一个元素为分界线（基准），将所有小于等于它的元素移到该元素的前面，将所有大于它的元素移到该元素的后面。



#include <stdio.h>

void partition(int *L*[], int *n*)

{

    int pivot = *L*[0];

    int temp;

    int left = 1; *// left指向第一个元素*

    int right = *n* - 1; *// right指向最后一个元素*

    while (left <= right)

    {

        while (left <= right && *L*[left] <= pivot)

        {

            left++;

        }

        while (left <= right && *L*[right] > pivot)

        {

            right--;

        }

        if (left < right)

        {

            temp = *L*[left];

*L*[left] = *L*[right];

*L*[right] = temp;

        }

    }

    pivot = *L*[right];

*L*[right] = *L*[0];

*L*[0] = pivot;

    if (*n* <= 1)

    {

        return;

    }

}

int main()

{

    int L[] = {50, 76, 20, 49, 8, 99, 64, 88, 34, 67};

    int n = sizeof(L) / sizeof(L[0]);

    partition(L, n);

    for (int i = 0; i < n; i++)

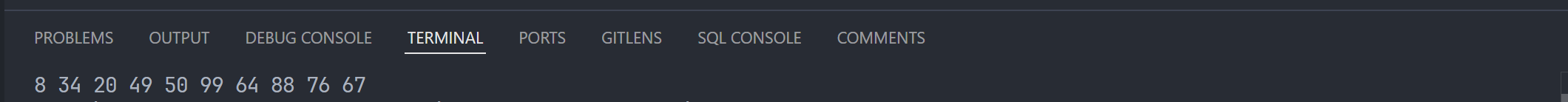
    {

        printf("%d ", L[i]);

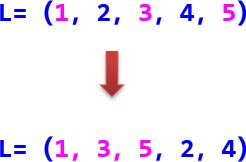
    }

    return 0;

}



Q2：有一个顺序表 *L*，假设元素类型 ElemType 为整型。设计一个尽可能高效的算法，将所有奇数移动到偶数的前面。



#include <stdio.h>

void range(int *L*[], int *n*)

{

    int left = 0;

    int right = *n* - 1;

    int temp;

    while (left < right)

    {

*// 左指针指向奇数，继续向右移动*

        while (left < right && *L*[left] % 2 != 0)

        {

            left++;

        }

*// 右指针指向偶数，继续向左移动*

        while (left < right && *L*[right] % 2 == 0)

        {

            right--;

        }

*// 交换左右指针指向的元素*

        if (left < right)

        {

            temp = *L*[left];

*L*[left] = *L*[right];

*L*[right] = temp;

        }

    }

}

int main()

{

    int L[] = {50, 76, 21, 49, 8, 99, 64, 88, 35, 67};

    int n = sizeof(L) / sizeof(L[0]);

    range(L, n);

    for (int i = 0; i < n; i++)

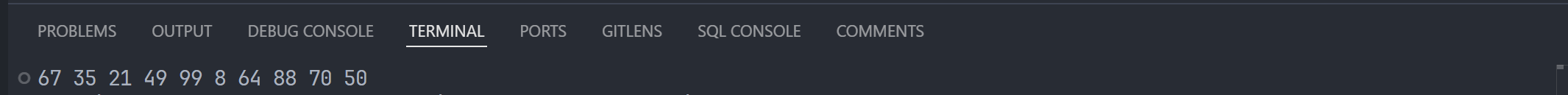
    {

        printf("%d ", L[i]);

    }

    return 0;

}



DDL: 06/12/2024