

Харківський національний університет імені В. Н. Каразіна Факультет
комп'ютерних наук
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ
Практична робота №4
дисципліна: «Алгоритми та структури даних»

Виконав: студент групи КС-22
Малютін Максим
Перевірив: викладач
Олешко Олег

Харків

Завдання. Напишіть програму для рішення головоломки.

Головоломка представляє собою масив цілих чисел, наприклад:

3 6 4 1 3 4 2 5 3 0

Зелений колір в першій клітинці вказує ваше теперішнє положення. На кожному етапі головоломки ви можете рухатися рівно на стільки клітинок, скільки вказано цілим числом в клітинці, в якій ви знаходитесь. Ви можете рухатися або вліво, або вправо, але не можете виходити за масив. Наприклад, єдиним дозволеним першим кроком є переміщення на три клітинки вправо, тому що немає місця для переміщення на три клітинки вліво. Мета головоломки - перемістити зелений маркер на 0 в останній клітинці. В наведеній конфігурації ви можете вирішити головоломку, зробивши такі кроки:

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

3 6 4 1 3 4 2 5 3 0

Незважаючи на те, що для наведеного прикладу ця головоломка може бути розв'язана — і, дійсно, має більше одного рішення, — деякі головоломки цього виду можуть не мати рішення взагалі, наприклад, наступна:

3 1 2 3 0

В цій головоломці ви будете рухатися між двома трійками, але не зможете потрапити до інших клітинок.

Пояснення до завдання

1. Всі числа в масиві є позитивними.
2. Змінювати значення в масиві заборонено.
3. Останній елемент масиву дорівнює 0. (Взагалі кажучи, якщо ви туди дісталися, не важливо, яке число там зберігається.)
4. Інших нулів в масиві немає. (Взагалі кажучи, якщо будуть інші нулі и ви потрапите в таку клітинку, ви в ній і застрягнете.)
5. Існує дозволений перехід із кожної клітинки масиву. (Уявіть, якщо в одній із клітинок вищевказаного прикладу буде число 20.)

Вимоги до програми

Програма заповнює масив розміром 10 випадковими значеннями (не більше 9), виводить на екран вихідний масив, а потім всі кроки рішення задачі та ВСІ варіанти рішення в вигляді:

3 R 1 R 3 L 6 R 5 L 4 R 2 R 3 L 4 R 0 (де вказується число в клітинці та напрям руху «R/L» — right/left)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define SIZE 10

// Структура для збереження кроку (значення і напрямки)
typedef struct {
    int value;
    char direction;
} Step;

// Функція для пошуку всіх шляхів
void solvePuzzle(int *arr, int size, int pos, Step *steps, int stepIndex, int *found) {
    // Перевірка виходу за межі масиву або зациклення
    if (pos < 0 || pos >= size || stepIndex >= size) {
        return;
    }

    if (arr[pos] == 0 && pos == size - 1) {
        // Дісталися до останньої клітинки
        for (int i = 0; i < stepIndex; i++) {
            printf("%d %c ", steps[i].value, steps[i].direction);
        }
        printf("0\n"); // Остання клітинка завжди 0
        *found = 1; // Встановлюємо, що рішення знайдено
        return;
    }

    // Якщо рішення вже знайдено, виходимо
    if (*found) return;

    // Додаємо поточний крок (вправо і вліво)
    if (pos + arr[pos] < size && !*found) {
```

```

        steps[stepIndex].value = arr[pos];
        steps[stepIndex].direction = 'R';
        solvePuzzle(arr, size, pos + arr[pos], steps, stepIndex + 1, found);
    }
    if (pos - arr[pos] >= 0 && !*found) {
        steps[stepIndex].value = arr[pos];
        steps[stepIndex].direction = 'L';
        solvePuzzle(arr, size, pos - arr[pos], steps, stepIndex + 1, found);
    }
}

int main() {
    srand(time(0)); // Ініціалізація генератора випадкових чисел

    int arr[SIZE];
    Step steps[SIZE]; // Масив для збереження кроків
    int found = 0; // Прапорець для позначення знайденого рішення

    // Заповнюємо масив випадковими значеннями (від 1 до 9), крім останнього
    // елемента
    for (int i = 0; i < SIZE - 1; i++) {
        arr[i] = rand() % 9 + 1; // Числа від 1 до 9
    }
    arr[SIZE - 1] = 0; // Останній елемент масиву — завжди 0

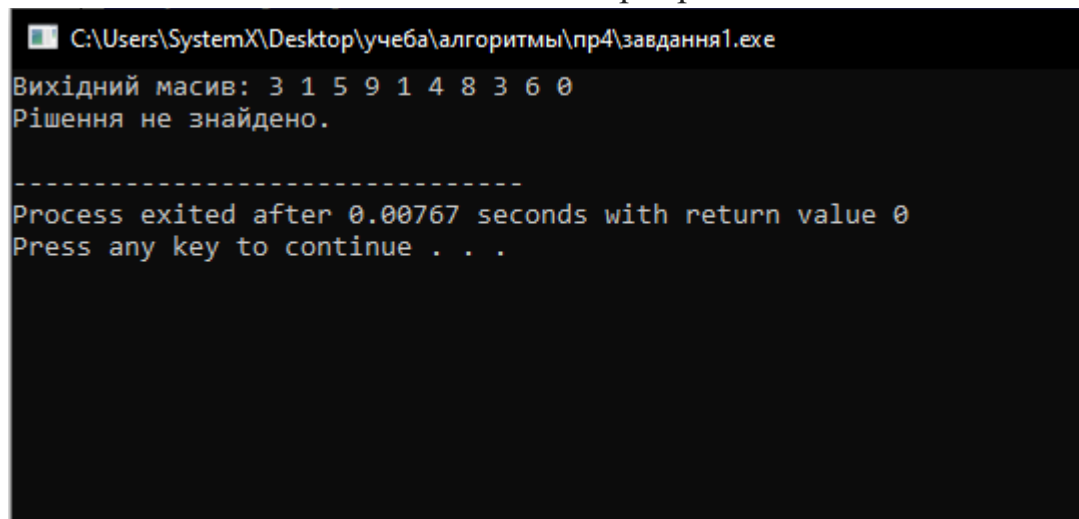
    // Виводимо вихідний масив
    printf("Вихідний масив: ");
    for (int i = 0; i < SIZE; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    // Шукаємо всі варіанти рішень
    solvePuzzle(arr, SIZE, 0, steps, 0, &found);
}

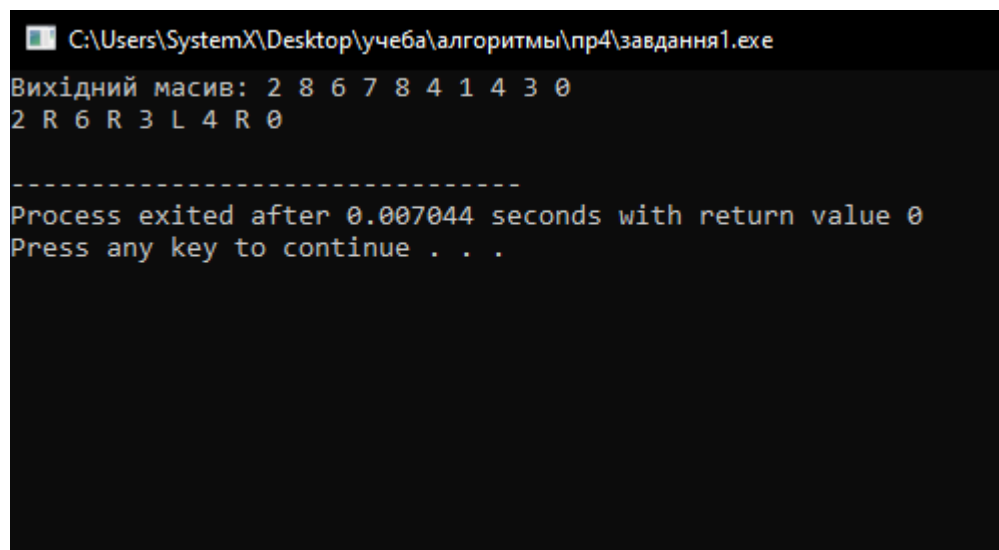
```

```
if (!found) {  
    printf("Рішення не знайдено.\n");  
}  
  
return 0;  
}
```

Лістинг 1 – код програми



```
C:\Users\SystemX\Desktop\учеба\алгоритмы\пр4\завдання1.exe  
Вихідний масив: 3 1 5 9 1 4 8 3 6 0  
Рішення не знайдено.  
-----  
Process exited after 0.00767 seconds with return value 0  
Press any key to continue . . .
```



```
C:\Users\SystemX\Desktop\учеба\алгоритмы\пр4\завдання1.exe  
Вихідний масив: 2 8 6 7 8 4 1 4 3 0  
2 R 6 R 3 L 4 R 0  
-----  
Process exited after 0.007044 seconds with return value 0  
Press any key to continue . . .
```

```
C:\Users\SystemX\Desktop\учеба\алгоритмы\пр4\завдання1.exe
Вихідний масив: 8 7 2 7 4 1 6 3 4 0
Рішення не знайдено.

-----
Process exited after 0.007706 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\SystemX\Desktop\учеба\алгоритмы\пр4\завдання1.exe
Вихідний масив: 8 8 1 6 6 5 3 6 7 0
8 R 7 L 8 R 0

-----
Process exited after 0.007273 seconds with return value 0
Press any key to continue . . .
```

Скріншот 1 – виконання програми