МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп’ютерних наук та штучного інтелекту

ПРАКТИЧНА РОБОТА №4

Виконав: студент 2 курсу групи КС22
Спеціальності 122 «Комп’ютерні науки»
Скрипняк Тарас Артемович
Прийняв: викладач
Олешко О.І.

Харків – 2024

Завдання: Напишіть програму для рішення головоломки.
Головоломка представляє собою массив 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, единим 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.

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

typedef struct Node {
    int data;
    struct Node *left;
    struct Node *right;
} Node;

Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->left = newNode->right = NULL;
    return newNode;
}

void fillBranch(int arr[], int *visited, int size, int index, Node* current) {
    if (current == NULL || !(current->data)) return;

    if (index + arr[index] < size && !visited[index + arr[index]]) {
        visited[index] = 1;
        current->right = createNode(arr[index + arr[index]]);
        fillBranch(arr, visited, size, index + arr[index], current->right);
        visited[index] = 0;
    }

    if (index - arr[index] >= 0 && !visited[index - arr[index]]) {
        visited[index] = 1;
        current->left = createNode(arr[index - arr[index]]);
        fillBranch(arr, visited, size, index - arr[index], current->left);
    }

}

void printPathsRecursive(Node* root, char path[], int pathLen, int dir) {
    if (root == NULL) return;

    if (dir == 0) path[pathLen++] = 'L';
    else if (dir == 1) path[pathLen++] = 'R';
    path[pathLen++] = root->data + '0';

    if (root->data == 0) {
        path[pathLen++] = '\0';
        printf("%s\n", path);
    } else {
        printPathsRecursive(root->left, path, pathLen, 0);
        printPathsRecursive(root->right, path, pathLen, 1);
    }
}

int main() {
    int arr[10];
    int visited[10] = { 1 };
    srand(time(NULL));

    for (int i = 0; i < 9; i++)
        arr[i] = rand() % 9 + 1;
    arr[9] = 0;
```

```
    printf("Generated array: ");
    for (int i = 0; i < 10; i++)
        printf("%d ", arr[i]);
        printf("\n");

    Node* root = createNode(arr[0]);
    fillBranch(arr, visited, 10, 0, root);

    char path[19];
    printPathsRecursive(root, path, 0, -1);

    return 0;
}
```

Лістинг

```
bouncytorch@AORUS:~/Repos/homework-c/pr4/tarik$ ./a.out
Generated array: 4 2 2 6 1 4 9 7 9 0
4R1L6R0
4R1R4L2R6R0
4R1R4R0
```

Результат виконання