Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2

з дисципліни
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-22
Тимофеєв Даниіл Костянтинович
номер у списку групи: 20

Перевірила:

Молчанова А. А.

Київ 2023

Постановка задачі :

1. Створити список з n (n>0) елементів (n вводиться з клавіатури), якщо інша кількість елементів не вказана у конкретному завданні.

2. Тип ключів (інформаційних полів) задано за варіантом.

3. Значення елементів списку взяти самостійно такими, щоб можна було продемонструвати коректність роботи алгоритму програми. Введення значень елементів списку можна виконати довільним способом (випадкові числа, формування значень за формулою, введення з файлу чи з клавіатури).

4. Вид списку (черга, стек, дек, прямий однозв'язний лінійний список, обернений однозв'язний лінійний список, двозв'язний лінійний список, однозв'язний кільцевий список, двозв'язний кільцевий список) вибрати самостійно з метою найбільш доцільного рішення поставленої за варіантом задачі.

5. Виконати над створеним списком дії, вказані за варіантом, та коректне звільнення пам'яті списку.

6. При виконанні заданих дій, виводі значень елементів та звільненні пам'яті списку вважати, що довжина списку (кількість елементів n чи 2n) невідома на момент виконання цих дій.

7. Повторювані частини алгоритму необхідно оформити у вигляді процедур або функцій (для створення, обробки, виведення та звільнення пам'яті списків) з передачею списку за допомогою параметра(ів)

Завдання для конкретного варіанту

Варіант 20

Ключами елементів списку є цілі числа. Кількість елементів списку повинна дорівнювати 2n. Перекомпонувати елементи списку так, розташування елементів було наступним: a1, an + 1, a2, an + 2, a3, ..., an, a2n, де ai – i-й компонент списку, не використовуючи додаткових структур даних, крім простих змінних (тобто «на тому ж місці»).

Текст програми :

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

typedef struct Node {
    int data;
```

```
    struct Node* next;
} Node;

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

Node* insertNode(Node* head, int data) {
  if (!head) {
    head = createNode(data);
  } else {
    Node* newNode = createNode(data);
    newNode->next = head;
    head = newNode;
  }
  return head;
}

void printList(Node* head) {
  Node* ptr = head;
  while (ptr) {
    printf("%d ", ptr->data);
    ptr = ptr->next;
  }
  printf("\n");
}

void rearrangeList(Node** headRef) {
  Node* head = *headRef;
  if (!head || !head->next)
    return;

  Node* slow = head;
  Node* fast = head->next;

  while (fast && fast->next) {
    slow = slow->next;
    fast = fast->next->next;
  }

  Node* current1 = head;
  Node* current2 = slow->next;
  slow->next = NULL;

  while (current2) {
    Node* temp1 = current1->next;
    Node* temp2 = current2->next;
    current1->next = current2;
    current2->next = temp1;
    current1 = temp1;
```

```
    current2 = temp2;
  }
}

void freeList(Node* head) {
  Node* temp = NULL;
  while (head != NULL) {
    temp = head;
    head = head->next;
    free(temp);
  }
}

int main() {
  int data;
  Node* head = NULL;
  printf("Enter the values for the nodes (enter -1 to stop):
\n");
  while (1) {
    printf("Enter a value: ");
    scanf("%d", &data);
    if (data == -1)
      break;
    head = insertNode(head, data);
  }
  printf("Original list: ");
  printList(head);
  rearrangeList(&head);
  printf("Rearranged list: ");
  printList(head);
  freeList(head);
  return 0;
}
```

Тестування програми

Тест 1 : n = 2. Послідовність така : a1 = 2; a2 =5; a3 = 6; a4 =7

Після перекомпонування послідовність буде такою: a1, a3, a2, a4

```
Enter the values for the nodes (enter -1 to stop):
Enter a value: 7
Enter a value: 6
Enter a value: 5
Enter a value: 2
Enter a value: -1
Original list: 2 5 6 7
Rearranged list: 2 6 5 7
```

Тест 2 : n = 5. Послідовність така : a1 = 1; a2 =2; a3 = 3; a4 =4; a5 =5; a6 = 6; a7 =7 ; a8 =8; a9 =9; a10 =10.

Після перекомпонування послідовність буде такою : a1, a6, a2, a7, a3, a8, a4, a9, a5 , a10 (1,6,2,7,3,8,4,9,5,10)

```
Enter the values for the nodes (enter -1 to stop):
Enter a value: 10
Enter a value: 9
Enter a value: 8
Enter a value: 7
Enter a value: 6
Enter a value: 5
Enter a value: 4
Enter a value: 3
Enter a value: 2
Enter a value: 1
Enter a value: -1
Original list: 1 2 3 4 5 6 7 8 9 10
Rearranged list: 1 6 2 7 3 8 4 9 5 10
```

Тест 3 : n = 6. Послідовність така : a1 = 3; a2 =8; a3 = 48; a4 =5; a5 =10; a6 = 11; a7 =21 ; a8 =9; a9 =34; a10 =45; a11 = 89; a12 = 64.

Після перекомпонування послідовність буде такою : a1, a7, a2, a8, a3, a9, a4, a10, a5, a11, a6, a12 (3,21,8,9,48,34,5,45,10,89,11,64)

```
Enter the values for the nodes (enter -1 to stop):
Enter a value: 64
Enter a value: 89
Enter a value: 45
Enter a value: 34
Enter a value: 9
Enter a value: 21
Enter a value: 11
Enter a value: 10
Enter a value: 5
Enter a value: 48
Enter a value: 8
Enter a value: 3
Enter a value: -1
Original list: 3 8 48 5 10 11 21 9 34 45 89 64
Rearranged list: 3 21 8 9 48 34 5 45 10 89 11 64
```