

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання

Лабораторних та практичних робіт № 4

з дисципліни: «Мови та парадигми програмування»

з розділу: «Рік 4. Прості структури даних. Одновимірні масиви. Двовимірні масиви. Алгоритми обробки»

Виконала:

студентка групи ШІ-14 - Гуменюк Тетяна В'ячеславівна

Львів 2023

Тема роботи:

Ознайомлення з типами простих структур даних одновимірними та двовимірними масивами, алгоритмами обробки.

Мета роботи:

Створення одновимірних та двовимірних масивів.

Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
 - Тема №1: Структури даних
 - Тема №2: Одновимірні масиви
 - Тема №3: Двовимірні масиви
 - Тема №4: Вказівники
 - Тема №5: Алгоритми обробки
- 2) Індивідуальний план опрацювання теорії:
 - Тема №1: Структури даних
 - Джерела Інформації:
 - <https://robotdreams.cc/uk/blog/58-structure-your-data-please>
 - https://financial.lnu.edu.ua/wp-content/uploads/2016/09/lektsiia_VF_4_tema-4_perehliad.pdf
 - Що опрацьовано:
 - Види структур даних
 - Статус: Ознайомлена
 - Початок опрацювання теми: 4 грудня
 - Звершення опрацювання теми: 4 грудня
 - Тема №2: Одновимірні масиви
 - Джерела Інформації:
 - <https://www.youtube.com/watch?v=ULdbOaMBPYc>
 - Що опрацьовано:
 - Створення одновимірного масиву
 - Статус: Ознайомлена
 - Початок опрацювання теми: 4 грудня
 - Звершення опрацювання теми: 4 грудня
 - Тема №3: Двовимірні масиви
 - Джерела Інформації:
 - <https://www.youtube.com/watch?v=2R1eYJHeFxI>
 - Що опрацьовано:
 - Створення багатовимірного масиву
 - Статус: Ознайомлена
 - Початок опрацювання теми: 4 грудня
 - Звершення опрацювання теми: 5 грудня
 - Тема №4: Вказівники
 - Джерела Інформації: ▪ <https://acode.com.ua/urok-84-vkazivnyky/>
 - <https://acode.com.ua/urok-97-vkazivnyky-na-vkazivnyky/>
 - Що опрацьовано:
 - Вказівник та вказівник на вказівник

- Статус: Ознайомлена
- Початок опрацювання теми: 5 грудня
- Звершення опрацювання теми: 10 грудня
- Тема №5: Алгоритми обробки
- Джерела Інформації:
 - <https://dan-it.com.ua/uk/blog/algoritmy-i-struktury-dannyh-dlja-nachinajushhih-preimushhestva-metodiki-izuchenija-i-poleznye-resursy/#i>
- Що опрацьовано: ▪ Ознайомлення з алгоритмом бінарного пошуку
- Статус: Ознайомлена
- Початок опрацювання теми: 4 грудня
- Звершення опрацювання теми: 4 грудня

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

- Завдання №1: VNS Lab 4 variant 24
- Деталі завдання: 24.

- 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).
- 2) Роздрукувати отриманий масив, починаючи з K -ого елемента й до $K-1$.
- 3) Впорядкувати елементи за зростанням
- 4) Знищити з кільця парні елементи.
- 5) Роздрукувати отриманий масив, починаючи з K -ого елемента і до $K-1$.

- Завдання №2: VNS Lab 5 variant 24
- Деталі завдання:

24. Визначити скільки елементів двовимірному масиву більші від будь-якого елемента на головній діагоналі.

- Завдання №3: Alotester Lab 2 variant 1
- Деталі завдання:

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елемента.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

- Завдання №4: Alotester Lab 3 variant 3
- Деталі завдання:

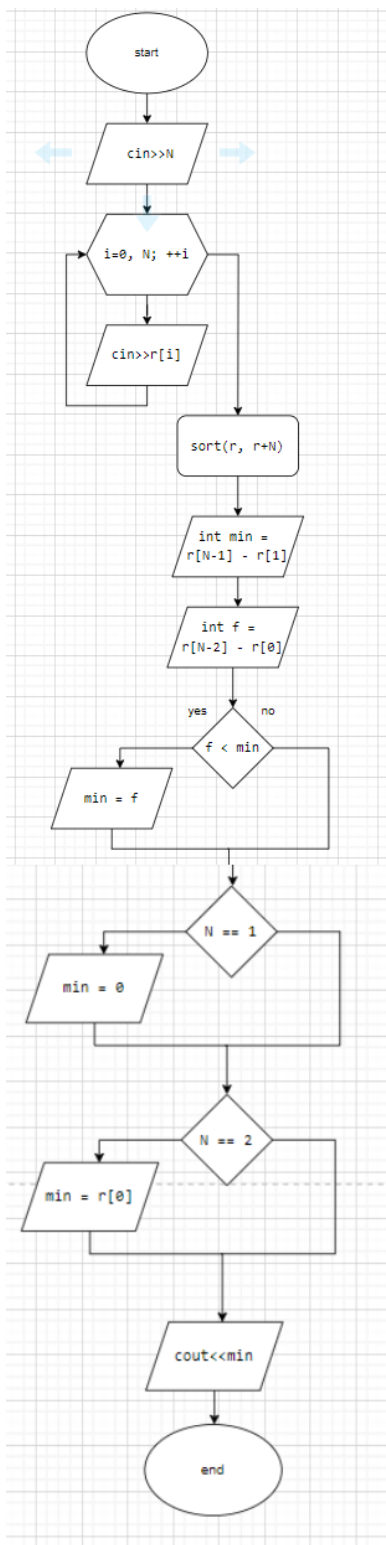
Вам дана стрічка s .

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

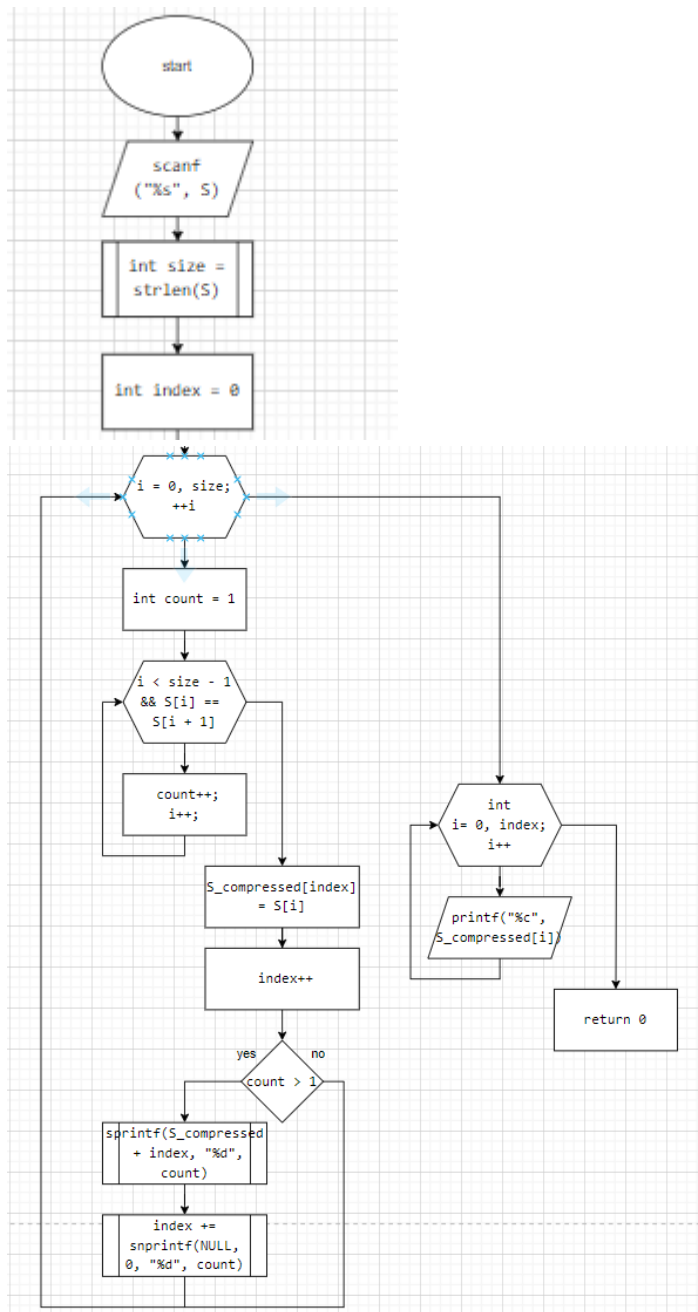
2. Дизайн та планована оцінка часу виконання завдань:

Програма №3: Alotester Lab 2 variant 1

- Блок-схема



- Планований час на реалізацію — 3 години
- Програма №4: Alotester Lab 3 variant 3
- Блок-схема



- Планований час на реалізацію – 4 години

3. Код програм:

- Завдання №1: VNS Lab 4 variant 24

```

#include <stdio.h>
#include <string.h>
#include <algorithm>
#include <iostream>
using namespace std;
void review(int *mas, int n, int K)
{
    K=K%n;
    for(int i=K-1; i<n; i++)
    {
        printf("%d ", mas[i]);
    }
    for(int i=0; i<K-1; i++)
    {
        printf("%d ", mas[i]);
    }
}

```

```

    }
}

void odd (int *mas, int n, int K)
{
    K=K%n;
    for(int i=K-1; i<n; i++)
    {
        if(mas[i]%2!=0)
        {
            printf("%d ", mas[i]);
        }
    }
    for(int i=0; i<K-1; i++)
    {
        if(mas[i]%2!=0)
        {
            printf("%d ", mas[i]);
        }
    }
}

int main()
{
    int K;
    cin>> K;
    int d[] = {1, 6, 3, 2, 5, 5};
    review(d, sizeof(d)/sizeof(d[0]), K);
    sort(d, d+sizeof(d)/sizeof(d[0]));
    cout<<"\nsort: ";
    review(d, sizeof(d)/sizeof(d[0]), K);
    cout<<"\nno even: ";
    odd (d, sizeof(d)/sizeof(d[0]), K);
}

```

- Завдання №2: VNS Lab 5 variant 24

```

#include <iostream>
using namespace std;
int countOfLargerElements(int **matrix, int n)
{
    int diagonal_max;
    diagonal_max = matrix[0][0];
    for(int i=0; i<n; i++)
    {
        if(matrix[i][i] > diagonal_max)
        {
            diagonal_max = matrix[i][i];
        }
    }
    int count = 0;
    for(int row=0; row<n; row++)
    {
        for(int column=0; column<n; column++)
        {
            if(matrix[row][column]>diagonal_max)
            {
                count++;
            }
        }
    }
    return count;
}

```

```

- }
- int main()
- {
-     int n;
-     cout<<"enter dimension of square matrix: ";
-     cin>>n;
-     cout<<"enter elements of matrix: \n";
-     int **matrix = new int *[n];
-     for(int i=0; i<n;i++)
-     {
-         matrix[i] = new int [n];
-     }
-     for(int row=0; row<n; row++)
-     {
-         for(int column=0; column<n; column++)
-         {
-             cin>>matrix[row][column];
-         }
-     }
-     cout<<countOfLargerElements(matrix, n);
- }

```

- Завдання №3: Alotester Lab 2 variant 1

```

- #include <iostream>
- #include <algorithm>
- using namespace std;
- int main()
- {
-     int N;
-     cin>>N;
-     int r[N];
-     for (int i=0; i<N; ++i){
-         cin>>r[i];
-     }
-     sort(r, r+N);
-     int min = r[N-1] - r[1];
-     int f = r[N-2] - r[0];
-     if(f < min){
-         min = f;
-     }
-     if(N == 1){
-         min = 0;
-     }
-     if(N == 2){
-         min = r[0];
-     }
-     cout<<min;
- }

```

- Завдання №4: Alotester Lab 3 variant 3

```

- #include <stdio.h>
- #include <string.h>
-
- int main() {
-     char S[100000];
-     scanf("%s", S);
-
-     int size = strlen(S);
-     char S_compressed[size * 2];
-
-     int index = 0;
-     for (int i = 0; i < size; ++i)
-     {

```

```

- int count = 1;
- while (i < size - 1 && S[i] == S[i + 1])
- {
-     count++;
-     i++;
- }
-
- S_compressed[index] = S[i];
- index++;
-
- if (count > 1)
- {
-     sprintf(S_compressed + index, "%d", count);
-     index += snprintf(NULL, 0, "%d", count);
- }
- }
- for (int i = 0; i < index; i++) {
-     printf("%c", S_compressed[i]);
- }
-
- return 0;
- }

```

4. Результати виконання завдань, тестування та фактично затрачених час:

Завдання №1: VNS Lab 4 variant 24

```
4
2 5 5 1 6 3
sort: 5 5 6 1 2 3
no even: 5 5 1 3
```

- Час затрат на выполнение задания: 3 години

Завдання №2: VNS Lab 5 variant 24

```
enter dimension of square matrix: 4
enter elements of matrix:
6 5 2 3
4 5 8 1
9 2 1 5
5 2 6 1
2
```

- Час затрат на выполнение задания: 4 години

Завдання №3: Alotester Lab 2 variant 1

```
5
3 2 5 1 7
4
```

- Час затрачений на виконання завдання: 2,5 години

Завдання №4: Alotester Lab 3 variant 3

AAACBBBBBBBBBBBB
A3CB12

- Час затрат на выполнение задания: 5 часов