

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ УПРАВЛІННЯ

ЗВІТ  
З ЛАБОРАТОРНОЇ РОБОТИ №5  
ЗА ТЕМОЮ «РОБОТА З РЯДКАМИ, СТРУКТУРАМИ ТА ФАЙЛАМИ»

Виконав студент  
групи КН-221в  
Шулюпов Єгор Русланович

Перевірив  
Асистент кафедри  
Ольховий Олексій Михайлович

ХАРКІВ 2021

## Хід роботи

### Завдання 1.1

Прочитати з клавіатури речення (масив символів) за допомогою функції `getline()`, перевірити, чи є воно паліндромом і вивести відповідне повідомлення. Рекомендація: під час введення не використовувати великих літер.

Паліндром – це речення, що однаково читається в обох напрямках (зліва направо та справа наліво), наприклад, "аби ріці риба". При цьому пропуски та розділові знаки не враховуються. В програмній реалізації їх доцільно видалити та здійснювати перевірку отриманого рядка.

```
#include <iostream>
#include <string>
using namespace std;

void polin(char* text, int size) {
    for (int i = 0; i < (size / 2); i++)
    {
        if (text[i] != text[size - i - 1])
        {
            cout << "Not polin" << endl;
            break;
        }
        if ((size / 2) - 1 == i)
            cout << "Polim !!" << endl;
    }
}

int main()
{
    setlocale(LC_ALL, "");
    char str[128];
    char str_2[128];

    cin.getline(str, 128);

    for (int i = 0; i < 128; i++)
    {
        int a = str[i];
        if ((a >= 65) && (a <= 90)) {
            str[i] += 32;
        }
    }

    int j = 0;
```

```
for (int i = 0; i < 128; i++)
{
    if (str[i] != '\0') {
        if (str[i] >= 'a' && str[i] <= 'z') {
            str_2[j] = str[i];
            j++;
        }
        else break;
    }
}

polin(str_2, j);
return 0;
}
```

## Тести

```
maDam Im ADam
Polim !!

C:\Users\esulu\Desktop\folders\1 курс\labs_op\лаб_5\ConsoleApplication1\Debug\ConsoleApplication1_1.exe (процесс 26152) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

```
qwertyewq
Not polin

C:\Users\esulu\Desktop\folders\1 курс\labs_op\лаб_5\ConsoleApplication1\Debug\ConsoleApplication1_1.exe (процесс 15028) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

## Завдання 1.2

Створити структуру для опису точки в тривимірному просторі.

Написати програму, яка обчислює відстань між двома точками.

Прочитати координати точок за допомогою функції `scanf()`. Вивести результат за допомогою функції `printf()`.  
`#include <iostream>`

```
#include "stdio.h"
#include <math.h>
#include <iostream>

struct xyzP {
    double x, y, z;
public:
    xyzP(double x, double y, double z)
        :x(x), y(y), z(z)
    {
    }
};

double lenght(xyzP& first, xyzP& second) {
    return sqrt((second.x - first.x)*(second.x - first.x) + (second.y - first.y)*(second.y - first.y) + (second.z - first.z)*(second.z - first.z));
}

int main() {
    setlocale(LC_ALL, "");
    double x, y, z, x1, y1, z1;
    scanf_s("%lf %lf %lf %lf %lf %lf", &x, &y, &z, &x1, &y1, &z1);
    xyzP frst(x, y, z);
    xyzP scnd(x1, y1, z1);

    printf("%lf", lenght(frst, scnd));
    return 0;
}
```

## Тести

```
1
1
1
2
2
2
1,732051
C:\Users\esulu\Desktop\folders\1 курс\labs_op\лаб_5\ConsoleApplication2\Debug\ConsoleApplication2.exe (процесс 30176) за
вершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

### Завдання 1.3

Написати програму, яка забезпечує файлове введення та виведення і включає індивідуальне завдання попередньої лабораторної роботи. Слід реалізувати такі дії:

- визначення константи (n) яка визначає кількість стовпців двовимірного масиву
- відкриття файлу для читання (файл повинен бути підготовлений за допомогою текстового редактора)
- читання цілих чисел до кінця файлу і зберігання їх у зв'язаному списку
- створення двовимірного масиву в динамічній пам'яті; кількість рядків повинна бути обчислена на основі кількості зчитаних з файлу значень та визначеної кількості стовпців
- заповнення двовимірного масиву рядок за рядком; відсутні елементи останнього рядка повинні бути заповнені нулями
- видалення елементів зв'язаного списку з динамічної пам'яті
- реалізація попереднього індивідуального завдання
- зберігання результатів в новому файлі

- видалення масивів операцією delete

```
• #include <iostream>
• #include <fstream>
• using namespace std;
•
• struct Link
• {
•     int data;
•     Link* next;
• };
•
•
•
• int main()
• {
•     setlocale(LC_ALL, "ru");
•     int m;
•     ifstream in("array.txt");
•     ofstream out("output.txt");
•
•
•     Link* first = 0;
•     Link* last = 0;
•     Link* link;
•     int d;
•     int count = 0;
•     while (in >> d)
•     {
•         count++;
•         link = new Link;
•         link->data = d;
•         link->next = 0;
•         if (last == 0)
•         {
•             first = link;
•         }
•         else
•         {
•             last->next = link;
•         }
•         last = link;
•     }
•
•     int n;
•     cout << "vvedite kolichestvo stobcov matricy: ";
•     cin >> n;
•
•     if (count % 2 != 0) {
•         m = count / n + 1;
•     }
•     else {
•         m = count / n;
•     }
•
•     double* arr_sum = new double[m];
•     int** arr = new int* [m];
•
•     for (int i = 0; i < m; i++)
•     {
```

```

•         arr[i] = new int[n];
•     }
•     link = first;
•     for (int i = 0; i < m; i++)
•     {
•         for (int j = 0; j < n; j++)
•         {
•             if (link != 0) {
•
•                 arr[i][j] = link->data;
•
•                 link = link->next;
•
•             }
•
•             else {
•
•                 arr[i][j] = 0;
•
•             }
•         }
•     }
•
•
•
•
•     for (int i = 0; i < m; i++)
•     {
•         double sum = 0;
•         for (int j = 0; j < n; j++)
•         {
•             if (arr[i][j] <= 0 && arr[i][j] % 2 != 0)
•                 arr[i][j] *= 2;
•             if (arr[i][j] > 0)
•                 sum += log10(arr[i][j]);
•         }
•         arr_sum[i] = sum;
•     }
•
•
•
•     for (int i = 0; i < m; i++)
•     {
•         for (int j = 0; j < n; j++)
•         {
•             out << arr[i][j] << " ";
•         }
•         if (arr_sum[i] == 0) {
•             out << "v stroke net polozhitel'nyh elementov" << endl;
•         }
•         else
•         {
•             out << "sum = " << arr_sum[i] << endl;
•         }
•     }
•
•     // delete
•
•
•     for (int i = 0; i < m; i++)
•     {
•         delete[] arr[i];
•     }

```

- 
- `delete[] arr;`
- 
- `while (first)`
- {
- `link = first;`
- `first = first->next;`
- `delete link;`
- }
- `delete[] arr_sum;`
- `return 0;`
- }  
• }

## Вправи для контролю

### Завдання 1

Прочитати рядок, замінити пропуски на символи підкреслення та вивести результат.

```
#include <iostream>
using namespace std;

int main() {
    setlocale(LC_ALL, "");
    cout << "введите строку(двойной пробел - конец строки): ";
    const int n = 128;
    char str[n];
    cin.getline(str, n);
    for (int i = 0; i < n; i++)
    {
        if ((str[i] == ' ') and (str[i + 1] == ' ')) {
            break;
        }
        if (str[i] == ' ') {
            str[i] = '_';
        }
        cout << str[i];
    }
    return 0;
}
```

Завдання 2

Визначити структуру для представлення двох цілих чисел, а потім створити і викликати функцію, яка отримує як аргумент структуру створеного типу і обчислює добуток елементів структури.

```
#include <iostream>
using namespace std;

struct num {
    double a, b;
    num(double a, double b)
        :a(a), b(b)
    {
    }
}
```



```
};

double multy(num& pare) {
    return pare.a * pare.b;
}

int main()
{
    setlocale(LC_ALL, "");
    double a, b;
    cout << "введите два числа: ";
    cin >> a >> b;
    num pare_1(a, b);
    cout << "произведение: " << multy(pare_1);
    return 0;
}
```

### Завдання 3

Написати програму, яка читає цілі до значення 0 і обчислює добуток цих чисел без останнього нульового значення.

```
#include <iostream>
using namespace std;
int main()
{
    int a; int m = 1;
    setlocale(LC_ALL, "");
    while(true){
        cin >> a;
        if (a != 0) {
            m *= a;
        }
        else {
            cout << "произведение равно: " << m;
            break;
        }
    }
    return 0;
}
```

### Завдання 4

Написати програму, яка читає цілі до кінця файлу і обчислює добуток ненульових значень.

```
#include <iostream>
#include <fstream>

using namespace std;
int main()
{
    ifstream in("input.txt");
    int a; int m = 1;
    setlocale(LC_ALL, "");
    while (in >> a) {
        if (a != 0) {
            m *= a;
        }
    }
}
```

```

    }
}

cout << "произведение равно: " << m;
return 0;
}

```

## Завдання 5

Написати програму, яка визначає масив чисел і записує в текстовий файл суми елементів з непарними індексами.

```

#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "");
    ofstream out("output.txt");
    cout << "vvedite kolichestvo elementov: " << endl;
    double n;
    cin >> n;
    double* arr = new double[n];
    double sum = 0;
    cout << "vvedite massive: " << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
    for (int i = 0; i < n; i++)
    {
        if (i % 2 != 0) {
            sum += arr[i];
            out << sum << " ";
        }
    }
    delete [] arr;
    return 0;
}

```

## Висновок:

Отже, в ході роботи я навчився реалізовувати різні структури на мові програмування C++ та виконувати роботу зі рядками та файлами.