

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

Кафедра дискретного аналізу

Операційні системи та системне програмування
Лабораторна робота №5

Виконав
Студент групи ПМІ-43
Заречанський Олексій
Викладач
Доц. Черняхівський Володимир

2023

1. Складені функції:

- Функція знаходження мінімального елементу `int` з `int*`, повертає `int`.
- Функція яка показує чи містить заданий `char*` інший заданий `char*`, з параметром `bool` який дозволяє перемикати case-sensitivity, повертає `bool`.
- Функція яка знаходить середнє значення `double*` та повертає його як `double`.
- Функція яка дозволяє порахувати кількість певного `char` у `char*`, з параметром `bool` який дозволяє перемикати case-sensitivity, повертає `int` - число входжень `char`.

2. Створюю проект в Visual Studio де реалізую ці функції, а також тестую їх.

Код функцій:

```
static int Minimum(int* selected, int arrSize)
{
    if (arrSize == 0) {
        return 0;
    }
    int min = selected[0];

    for (int i = 1; i < arrSize / sizeof(min); i++) {
        if (selected[i] < min) {
            min = selected[i];
        }
    }
    return min;
}
```

```

static bool Contains(char* toCheck, int checkedSize, char* toFind, int foundSize, bool caseSensitive) {
    int checkedLen = checkedSize - 1;
    int foundLen = foundSize - 1;

    for (int i = 0; i < checkedLen; i++) {

        if (checkedLen < foundLen + i) {
            return false;
        }

        bool result = true;

        for (int j = 0; j < foundLen; j++) {
            char left = toCheck[i + j];
            char right = toFind[j];

            if (!caseSensitive) {
                left = tolower(left);
                right = tolower(right);
            }

            if (left != right) {
                result = false;
                break;
            }
        }

        if (result) {
            return true;
        }
    }

    return false;
}

```

```

static double Average(int* numArr, int arrSize) {
    if (arrSize == 0) {
        return 0;
    }

    int arrLen = arrSize / sizeof(int);
    double sum = 0;

    for (int i = 0; i < arrLen; i++) {
        sum += numArr[i];
    }

    return sum / arrLen;
}

```

```

static int Count(char* text, int textSize, char counted, bool caseSensitive) {
    int counter = 0;

    int textLen = textSize - 1;

    char toFind = caseSensitive ? counted : tolower(counted);

    for (int i = 0; i < textLen; i++) {
        char tested = caseSensitive ? text[i] : tolower(text[i]);

        if (tested == toFind) {
            counter++;
        }
    }

    return counter;
}

```

Тести:

```
int findMin[5] = {3, 1, -5, 6, 0};
int min = Functions::Minimum(findMin, sizeof(findMin));

if (min != -5) {
    cout << "Error in Minimum function, result has to be -5, not " << min << endl;
}
else {
    cout << "Minimum funciton works correctly" << endl;
}

char text[] = "SoMe text";
char find[] = "ome";

bool resultContains = Functions::Contains(text, sizeof(text), find, sizeof(find), false);

if (resultContains) {
    cout << "Contains funciton works correctly with no regard for case" << endl;
}
else {
    cout << "Error in Contains function when case is not taken into account, result has to be true, not " << resultContains << endl;
}

bool resultContains2 = Functions::Contains(text, sizeof(text), find, sizeof(find), true);

if (!resultContains2) {
    cout << "Contains funciton works correctly with case sensitive values" << endl;
}
else {
    cout << "Error in Contains function when case is taken into account, result has to be false, not " << resultContains << endl;
}

int toBeAveraged[4] = { 8, 5, 4, 2 }; //average is 4.75
double average = Functions::Average(toBeAveraged, sizeof(toBeAveraged));
if (average == 4.75) {
    cout << "Average funciton works correctly" << endl;
}
else {
    cout << "Average funciton works incorrectly, result has to be 4.75, not " << average << endl;
}

char text2[] = "Occurrences of letter o will be counted here"; // o is here 2 times for lower case and 4 times for case insensitive
char letter = 'o';
int occurrences1 = Functions::Count(text2, sizeof(text2), letter, true);

if (occurrences1 == 2) {
    cout << "Count funciton works correctly with case sensitive values" << endl;
}
else {
    cout << "Error in Count function when case is taken into account, result has to be 2, not " << occurrences1 << endl;
}

int occurrences2 = Functions::Count(text2, sizeof(text2), letter, false);

if (occurrences2 == 4) {
    cout << "Count funciton works correctly with case insensitive values" << endl;
}
else {
    cout << "Error in Count function when case is not taken into account, result has to be 4, not " << occurrences2 << endl;
}
```

Як видно всі тести пройдено успішно.

```
Microsoft Visual Studio Debug Console

Minimum funciton works correctly
Contains funciton works correctly with no regard for case
Contains funciton works correctly with case sensitive values
Average funciton works correctly
Count funciton works correctly with case sensitive values
Count funciton works correctly with case insensitive values
```

3. Створюю такий самий проєкт з тими ж функціями, але заміню параметри функцій на типи Windows.

Функції:

```

static INT Minimum(LPINT selected, INT arrSize)
{
    if (arrSize == 0) {
        return 0;
    }
    INT min = selected[0];

    for (INT i = 1; i < arrSize / sizeof(min); i++) {
        if (selected[i] < min) {
            min = selected[i];
        }
    }

    return min;
}

```

```

static BOOL Contains(LPTSTR toCheck, INT checkedSize, LPTSTR toFind, INT foundSize, BOOL caseSensitive) {
    INT checkedLen = checkedSize - 1;
    INT foundLen = foundSize - 1;

    for (INT i = 0; i < checkedLen; i++) {
        if (checkedLen < foundLen + i) {
            return false;
        }

        BOOL result = true;

        for (INT j = 0; j < foundLen; j++) {
            CHAR left = toCheck[i + j];
            CHAR right = toFind[j];

            if (!caseSensitive) {
                left = tolower(left);
                right = tolower(right);
            }

            if (left != right) {
                result = false;
                break;
            }
        }

        if (result) {
            return true;
        }
    }

    return false;
}

```

```

static DOUBLE Average(LPINT numArr, INT arrSize) {
    if (arrSize == 0) {
        return 0;
    }

    INT arrLen = arrSize / sizeof(INT);
    DOUBLE sum = 0;

    for (INT i = 0; i < arrLen; i++) {
        sum += numArr[i];
    }

    return sum / arrLen;
}

```

```
static INT Count(LPTSTR text, INT textSize, CHAR counted, BOOL caseSensitive) {
    INT counter = 0;

    INT textLen = textSize - 1;

    CHAR toFind = caseSensitive ? counted : tolower(counted);







    for (INT i = 0; i < textLen; i++) {
        CHAR tested = caseSensitive ? text[i] : tolower(text[i]);

        if (tested == toFind) {
            counter++;
        }
    }

    return counter;
}
```

Тести не змінюю, вони повинні працювати без змін.

При спробі запуску отримую помилку

	Code	Description
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"
	E0167	argument of type "char *" is incompatible with parameter of type "LPTSTR"

Заміню в налаштування проекту набір символів на мультібайтовий

Character Set	Use Multi-Byte Character Set
Whole Program Optimization	Not Set
MSVC Toolset Version	Use Unicode Character Set
✓ C++/CLI Properties	Use Multi-Byte Character Set
Common Language Runtime Support	<inherit from parent or project defaults>
NET Target Framework Version	

Після цього все працює коректно

```
Microsoft Visual Studio Debug Console

Minimum funciton works correctly
Contains funciton works correctly with no regard for case
Contains funciton works correctly with case sensitive values
Average funciton works correctly
Count funciton works correctly with case sensitive values
Count funciton works correctly with case insensitive values
```

4. Створюю новий DLL проект і перенесу туди функції, напишу їх прототипи та експортую їх.

```
OS&SPSDDL (Global Scope) Contains(LPTSTR toCheck, INT checkedSize,
1 // dllmain.cpp : Defines the entry point for the DLL application.
2 #include "pch.h"
3
4 extern "C" __declspec(dllexport) INT(__stdcall Minimum) (LPINT selected, INT arrSize);
5 extern "C" __declspec(dllexport) BOOL(__stdcall Contains) (LPTSTR toCheck, INT checkedSize, LPTSTR toFind, INT foundSize, BO
6 extern "C" __declspec(dllexport) DOUBLE(__stdcall Average) (LPINT numArr, INT arrSize);
7 extern "C" __declspec(dllexport) INT(__stdcall Count) (LPTSTR text, INT textSize, CHAR counted, BOOL caseSensitive);
8
9 INT Minimum(LPINT selected, INT arrSize)
10 {
11     if (arrSize == 0) {
12         return 0;
```

З якоїсь причини не знаходить тип Windows DOUBLE

```
4 DOUBLE Average(LPINT numArr, INT arrSize) {
5     if (arrSize == 0) {
6         return 0;
7     }
8
9     INT arrLen = arrSize / sizeof(INT);
10    DOUBLE sum = 0;
11
12    for (INT i = 0; i < arrLen; i++) {
13        sum += numArr[i];
14    }
15
16    return sum / arrLen;
17 }
```

Подивлюсь в попередньому проекті його місцезнаходження

```
static DOUBLE Average(LPINT numArr, INT arrSize) {
149
150 typedef double DOUBLE;
151
152 #ifndef _DWORDLONG_
153 typedef unsigned __int64 DWORDLONG;
154
155 typedef DWORDLONG *PDWORDLONG;
156
157 #endif // !_DWORDLONG_
158 #ifndef _ULONGLONG_
159 typedef __int64 ULONGLONG;
160
161 typedef unsigned __int64 ULONGLONG;
162
163 typedef ULONGLONG *PULONGLONG;
164
165 // ...
166
167 if (arrSize == 0) {
```

Хоч він і так був в зовнішніх референсах проекту, його все одно не знаходило, тому включу його вручну в файл.

```
// dllmain.cpp : Defines the e
#include "pch.h"
#include "WTypesbase.h"
```

Це вирішило проблему

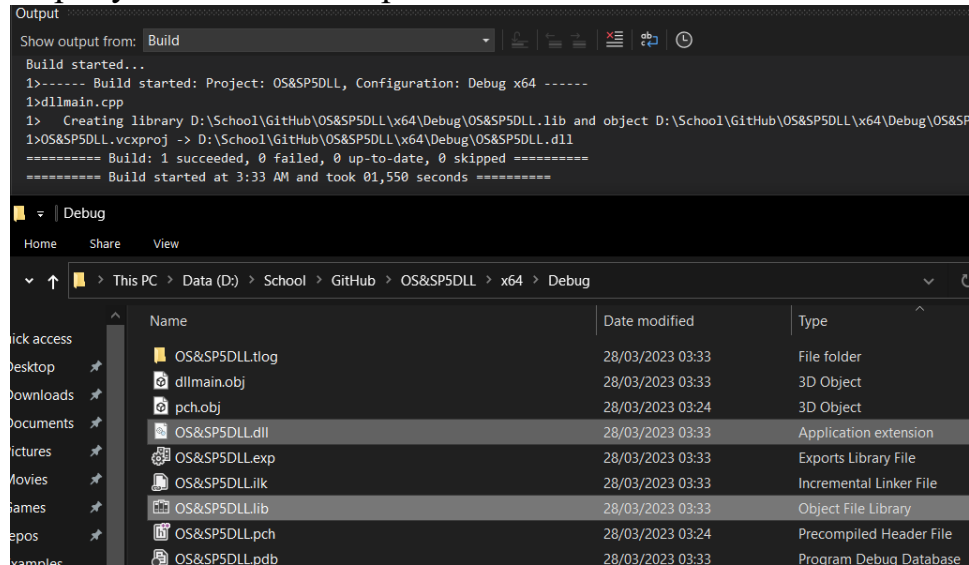
```

DOUBLE Average(LPINT numArr, INT arrSize) {
    if (arrSize == 0) {
        return 0;
    }

    INT arrLen = arrSize / sizeof(INT);
    DOUBLE sum = 0;

```

Спробуємо забілдити проект



Як видно все пройшло успішно і було створені файли проекту, зокрема файл OS&SP5DLL.dll та файл OS&SP5DLL.lib, які будуть використані для інших робіт та будуть включені з цим звітом як результат.