



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3

по дисциплине

«Структуры и алгоритмы обработки данных»

Тема: «Поразрядные операции и их применение»

Выполнил студент группы ИКБО-18-22

Ракитин В.А.

Принял преподаватель

Филатов А.С.

Лабораторная работа выполнена

«__»_____202__ г.

(подпись студента)

«Зачтено»

«__»_____202__ г.

(подпись руководителя)

Москва 2023

1. Цель работы

Получение навыков в разработке алгоритмов обработки текста – извлечение отдельных элементов. Получение навыков использования средств языка Си и C++ для реализации алгоритмов обработки текстовых данных.

2. Постановка задачи

1. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста нуль терминальную строку и средства языка C для выполнения операций над этой строкой.
2. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста строку `string` стандартной библиотеки шаблонов и возможности класса для выполнения действий со строкой.
3. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Вариант №21. Условие задания:

| | |
|------------|--|
| Упражнение | Дано предложение, состоящее из слов, разделенных пробелами. Среди слов могут быть числа шестнадцатеричного кода, которые начинаются с символа \$. Сформировать массив из десятичных значений шестнадцатеричных чисел, а сами шестнадцатеричные числа удалить из предложения. |
|------------|--|

3. Решение

`String` – это класс, который содержит массив символов, но автоматически управляет им для вас. Класс `string` был введен как альтернативный вариант для работы со строками типа `char*`. Строки C++ могут содержать встроенные символы `\0`, знать их длину без подсчета, быстрее, чем массивы символов,

выделенные в куче, для коротких текстов и защищают вас от переполнения буфера. Кроме того, они более удобочитаемы и просты в использовании.

Для решения упражнения была написана функция `char_symbol`. В этой функции создан массив, содержащий все символы, которые могут быть в 16-ой системе счисления. На вход функция получает символ, являющийся числом в 16-ой системе счисления. Далее функция возвращает это число в 10-ой системе счисления.

```
int char_symbol(char in) { //Функция, в которой находятся все возможные варианты числа в 16-ой с.с.  
    char out[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };  
    for (int i = 0; i < strlen(out); i++)  
    {  
        if (out[i] == in) {  
            return i; //возвращаем это число в 10-ой с.с.  
        }  
    }  
}
```

При запуске программы пользователь видит пользовательское меню, где необходимо установить количество символов в строке. Далее пользователь вводит строку.

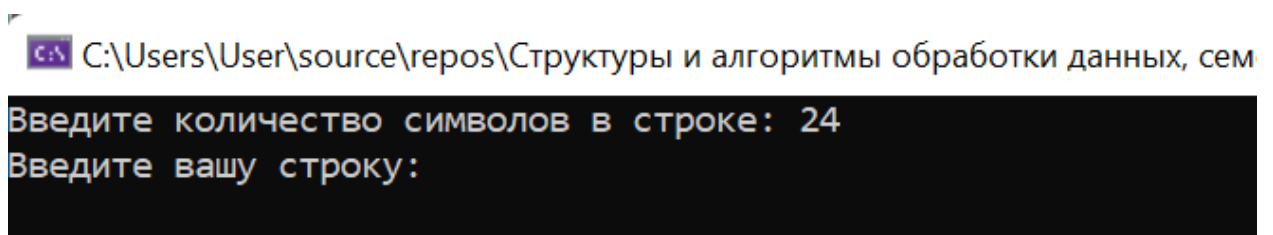
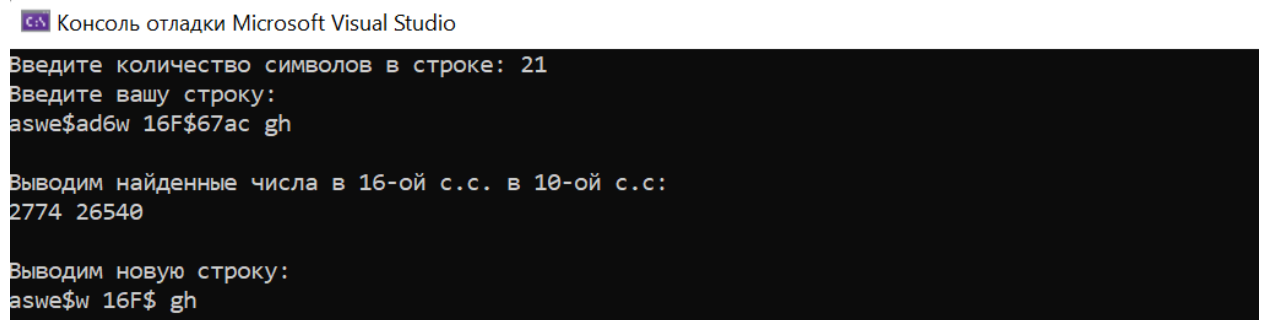


Рисунок 1. Интерфейс программы

4. Тестирование

Протестируем программой выполнение упражнения. Установим размер строки 21 и введём строку, состоящую из 21 символа. Наша строка: `aswe$adbw 16F$67ac gh`. В нашей строке нашлось два числа в 16-ой системе счисления, стоящие после символа `$`: `adb16` и `67ac16`. Переводим эти числа в 10-ую систему счисления: $adb_{16} = 10 * 16^2 + 13 * 16^1 + 6 * 16^0 = 2560 + 208 + 6$

$= 2774_{10}$; $67ac_{16} = 6 * 16^3 + 7 * 16^2 + 10 * 16^1 + 12 * 16^0 = 24576 + 1792 + 160 + 12 = 26540_{10}$. На рисунке 2 видно, что программа дала верный результат.



```
Консоль отладки Microsoft Visual Studio
Введите количество символов в строке: 21
Введите вашу строку:
aswe$ad6w 16F$67ac gh

Выводим найденные числа в 16-ой с.с. в 10-ой с.с:
2774 26540

Выводим новую строку:
aswe$w 16F$ gh
```

Рисунок 2. Решение упражнения

5. Вывод

В результате выполнения работы я:

1. Получил навыки в разработке алгоритмов обработки текста – извлечение отдельных элементов.
2. Получил навыки использования средств языка Си и C++ для реализации алгоритмов обработки текстовых данных.

6. Исходный код программы

```
#include<iostream> //Библиотека для ввода/вывода в консоль
using namespace std; //Пространство имён std

int char_symbol(char in) { //Функция, в которой находятся все возможные
    варианты числа в 16-ой с.с.
    char out[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
        'b', 'c', 'd', 'e', 'f' };
    for (int i = 0; i < strlen(out); i++)
    {
        if (out[i] == in) {
            return i; //возвращаем это число в 10-ой с.с.
        }
    }
}

int main() {
    setlocale(LC_ALL, "Rus"); // Установка русского языка для вывода в
    консоль
    /* узнаем размер массива */
    int i, n, q;
    char c;
```

```

printf("Введите количество символов в строке: ");
scanf_s("%d", &n);
c = getchar();

/* выделяем массив необходимого размера */
if (n > 0) {
    q = n + 1;
}
else {
    return -1;
} // кривой размер массива
char* stroka = (char*)malloc(sizeof(char) * q);
if (!stroka) {
    return -1; // памяти нет, выходим
}

/* записываем строку символов в массив */
i = 0;
q = 1;
printf("Введите вашу строку:\n");
while (q > 0)
{
    c = getchar();
    if (c != '\n' && i < n) {
        stroka[i++] = c;
    }
    else {
        stroka[i] = q = 0;
    }
}

cout << endl << "Выводим найденные числа в 16-ой с.с. в 10-ой с.с." <<
endl;

int sumk = 0;
for (int i = 0; i < strlen(stroka) - sumk; i++) {
    if (stroka[i] == '$') {
        int Sum = 0;
        int k = 0;
        int kk = 0;
        for (int j = i + 1; j < strlen(stroka); j++) {
            if ((stroka[j] == 'a') or (stroka[j] == 'b') or
(stroka[j] == 'c') or (stroka[j] == 'd') or
(stroka[j] == 'e') or (stroka[j] == 'f') or
(stroka[j] == 'o') or (stroka[j] == '1') or
(stroka[j] == '2') or (stroka[j] == '3') or
(stroka[j] == '4') or (stroka[j] == '5') or
(stroka[j] == '6') or (stroka[j] == '7') or
(stroka[j] == '8') or (stroka[j] == '9')) {
                /*Проверка, что символ является числом в 16-ой
с.с.*/
                k += 1;
                kk += 1;
                sumk += 1;
            }
            else {
                int ss = j - k;
                k -= 1;
                while (k != -1) {
                    Sum += char_symbol(stroka[ss]) * pow(16,
k); //Переводим число в 10-ую с.с.
                    k -= 1;
                    ss += 1;
                }
            }
        }
    }
}

```

```

        cout << Sum << " ";
        break;
    }
    }
    for (int j = i + 1; j < strlen(stroka) - kk; j++) {
        stroka[j] = stroka[j + kk];
    }
    (char*)realloc(stroka, (q-sumk)*sizeof(char));
}
cout << endl << endl << "Выводим новую строку:" << endl;
for (int i = 0; i < strlen(stroka) - sumk; i++) {
    cout << stroka[i];
}
cout << endl;
free(stroka); //Удаляем массив
return 0;
}

```

Таблица 1. Код программы с помощью массива элементов типа char

```

#include<iostream> //Библиотека для ввода/вывода в консоль
#include<string> //Библиотека string, предназначенного для работы с массивом
СИМВОЛОВ.
using namespace std; //Пространство имён std

int char_symbol(char in) { //Пространство имён std
    char out[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'b', 'c', 'd', 'e', 'f' };
    for (int i = 0; i < strlen(out); i++)
    {
        if (out[i] == in) {
            return i; //возвращаем это число в 10-ой с.с.
        }
    }
}

int main() {
    setlocale(LC_ALL, "Rus"); // Установка русского языка для вывода в
консоль
    string stroka;
    cout << "Введите строку" << endl;
    getline(cin, stroka);
    int sumk = 0;
    cout << endl << "Выводим найденные числа в 16-ой с.с. в 10-ой с.с:" <<
endl;
    for (int i = 0; i < stroka.length(); i++) {
        if (stroka[i] == '$') {
            int Sum = 0;
            int k = 0;
            int kk = 0;
            for (int j = i + 1; j < stroka.length(); j++) {
                if ((stroka[j] == 'a') or (stroka[j] == 'b') or
(stroka[j] == 'c') or (stroka[j] == 'd') or
(stroka[j] == 'e') or (stroka[j] == 'f') or
(stroka[j] == 'o') or (stroka[j] == '1') or
(stroka[j] == '2') or (stroka[j] == '3') or
(stroka[j] == '4') or (stroka[j] == '5') or
(stroka[j] == '6') or (stroka[j] == '7') or
(stroka[j] == '8') or (stroka[j] == '9')) {
                    /*Проверка, что символ является числом в 16-ой
с.с.*/
                    k += 1;

```

```

        kk += 1;
        sumk += 1;
    }
    else {
        int ss = j - k;
        k -= 1;
        while (k != -1) {
            Sum += char_symbol(stroka[ss]) * pow(16,
k); //Переводим число в 10-ую с.с.
            k -= 1;
            ss += 1;
        }
        cout << Sum << " ";
        break;
    }
}
for (int j = i + 1; j < stroka.length() - kk; j++) {
    stroka[j] = stroka[j + kk];
}
}
}
cout << endl << endl << "Выводим новую строку:" << endl;
for (int i = 0; i < stroka.length() - sumk; i++) {
    cout << stroka[i];
}
cout << endl;
return 0;
}

```

Таблица 2. Код программы с помощью метода string