ЛАБОРАТОРНАЯ РАБОТА №4

Цель работы:

Приобрести умения и практические навыки по работе со строками.

Теоретическая часть:

Строка – массив символов, который заканчивается нуль-символом (‘\0’). По его положению определяется фактическая длина строки.

Синтаксис объявления строк:

```
char имя [размерность];
```

Фактически длина строки на 1 меньше указанной размерности.

Строки при объявлении можно инициализировать строковыми константами:

```
char s[10] = “abc”;
```

Если размер массива больше размера строки, то оставшееся место в массиве заполняется нулевыми байтами (нуль-символом). Если строка при объявлении инициализируется, то размерность можно опускать, при этом компилятор сам выделяет нужное количество байт. Данные действия схожи при работе с массивами.

Строки, как и массивы, могут быть динамическими. Имя строки – указатель на первый элемент строки. При размещении строки в динамической памяти имени строки присваивается адрес выделенного участка памяти. Динамические строки нельзя инициализировать при создании.

Функции для работы со строками и символами содержатся в библиотеке <*string.h*>. Содержимое строк можно копировать из одной в другую, исследовать на содержание символов из одной строки в другой и т.д.

Ввод и вывод строк реализуется с помощью функций scanf_s и printf. При работе с этими функциями применяется спецификация %s. Ввод строки выполняется до первого пробельного символа. Прочитанные символы умещаются в строку, на конец которой ставится нуль-символ. Между % и s можно указать модификатор максимальной длины поля – целое число:

```
char s[10];
scanf_s(”%10s”, s); // в строку будет прочитано не более 10 символов
```

Также для ввода-вывода могут применяться функции gets и puts.

```
char * gets (char * s);
```

Эта функция считывает символы из входного потока до появления символа новой строки и помещает их в строку s, завершая ее нуль-символом. Возвращается указатель на строку, в случае возникновения ошибки – NULL.

```
int puts (char * s);
```

Эта функция реализует вывод строки на экран, завершающий нуль-символ при этом заменяется на символ новой строки. При успешном выводе возвращается неотрицательное значение, ошибка – *EoF*.

Для работы со строками предназначены функции, которые находятся в заголовочных файлах *string.h* и *cstring.h.*

Функция	**Пояснение**
Копирование строк	
*char * strcpy (char * S1, char * S2);*	Копирует S2 в S1 и возвращает S1. Копирование до момента, пока не встретится нуль-символ
*char *strncpy(char * S1, const char * S2, size_t n);*	Копирует n символов из строки S2 в S1. После копирования вручную нужно устанавливать нуль-символ
Длина строки	
*Size_t strlen (char * S);*	Возвращает длину строки (без учета символа завершения строки)
Конкатенация строк	
*char * strcat (char * S1, char * S2)*	Добавляет S2 к S1 и возвращает S1. В конец результирующей строки добавляется нуль - символ. При этом первая строка должна иметь достаточно свободного места, чтобы вместить вторую

*char *strncat(char * S1, const char * S2, size_t n);*	Дописывает не более n начальных символов строки S2 (или всю S2, если ее длина меньше) в конец S1. Результат сохраняется в S1
Поиск символа в строке	
*char * strchr (char * S, int ch);*	Возвращает указатель на первое вхождение символа ch в строку S, если его в строке нет, то возвращается NULL
*char *strrchr(const char *S, int c)*	То же, что strchr, но находит последнее вхождение символа c в строку S
Поиск подстроки в строке	
*char * strstr (char * S1, char * S2);*	Возвращает указатель на элемент из S1, с которого начинается S2 или NULL, если элемент не найден.
Сравнение строк	
*int strcmp(const char *S1, const char*S2);*	Сравнивает две строки. Возвращает целое меньше нуля, если S1 < S2, равное нулю, если S1 == S2, и большее нуля, если S1 > S2. С учётом регистра
*int stricmp(const char *S1, const char*S2);*	Сравнивает строку S1 со строкой S2 и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если S1 < S2, <0 – если S1 > S2. Без учёта регистра
*int strncmp(const char *S1, const char*S2, size_t n);*	Сравнивает n символов строки S1 со строкой S2 и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если S1 < S2, <0 – если S1 > S2. С учётом регистра

*int strnicmp(const char *S1, const char*S2, size_t n);*	Сравнивает n символов строки S1 со строкой S2 и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если S1 < S2, <0 – если S1 > S2. Без учёта регистра
Функции поиска	
*size_t strspn(const char *S1, const char *S2);*	Возвращает длину начального сегмента строки S1, содержащего только те символы, которые входят в строку S2
*size_t strcspn(const char *s, const char *reject);*	Возвращает длину начального сегмента строки S1, содержащего только те символы, которые не входят в строку S2
*char *strpbrk(const char *S1, const char *S2)*	Возвращает указатель первого вхождения любого символа строки S2 в строке S1
*char *strtok(char *S1, const char *S2);*	Сканирует первую строку в поисках первого участка, не содержащего символов из S2. Первый вызов функции возвращает указатель на начало первого участка и записывает 0 в S1 сразу после конца участка. Последующие вызовы с 0 в качестве 1-го аргумента обрабатывают строку дальше, пока еще есть такие участки. Если их нет, возвращается 0. Функцию применяют для выделения слов из предложения S1. В строке S2 находятся символы-разделители
Функции преобразования <stdlib.h>	
*double atof (const char *s);*	Преобразует строку S в тип double, если строка содержит недопустимое значение, то возвращаемое значение не определено

*int atoi (const char *S);*	Преобразует строку S в значение типа int, если строка содержит недопустимое значение, то возвращаемое значение не определено
*long int atol (const char *S)*	Преобразует строку S в тип long int
Обработка символов <ctype.h>	
isalnum (c)	Возвращает значение true, если с является буквой или цифрой, и false в других случаях
isalpha (c)	Возвращает значение true, если с является буквой, и false в других случаях
isdigit (c)	Возвращает значение true, если с является цифрой, и false в других случаях
islower (c)	Возвращает значение true, если с является буквой нижнего регистра, и false в других случаях
isupper (c)	Возвращает значение true, если с является буквой верхнего регистра, и false в других случаях
isspace (c)	Возвращает значение true, если с является пробелом, и false в других случаях
toupper (c)	Если символ с, является символом нижнего регистра, то функция возвращает преобразованный символ с в верхнем регистре, иначе символ возвращается без изменений.
tolower (c)	Если символ с, является символом верхнего регистра, то функция возвращает преобразованный символ с в нижнем регистре, иначе символ возвращается без изменений.

Задание 1. Написать собственный вариант стандартной функции. Сравнить результаты выполнения собственной функции и стандартной функции.

1. strchr;
2. strrchr.
3. strstr;
4. strlen;
5. strcmp;
6. stricmp;
7. strncmp;
8. strnicmp;
9. strcat;
10. strncat;
11. strcpy;
12. strncpy;

Задание 2. Дана строка длиной 80 символов. Словом называется последовательность непробельных символов, окруженных пробелами. Написать функцию, которая выполняет действия в соответствии с вариантом.

1. Вставляет слово «номер» перед словами, состоящими из цифр.
2. Переносит первое слова в конец предложения.
3. Переносит последнее слово в начало предложения.
4. Удаляет все слова в предложении, кроме первого и последнего.
5. Удаляет все символы между скобками.
6. Все слова, которые начинаются с буквы «а», переводит в верхний регистр.
7. Вычисляет сумму чисел, которые встречаются в предложении.
8. Находит самое длинное слово в предложении.
9. Находит самое короткое слово в предложении.
10. Все слова, которые заканчиваются на букву «а», начинает с заглавной буквы.
11. Выводит на экран все слова, содержащие цифры.
12. Выводит на экран все слова, не содержащие цифры.
13. Подсчитывает количество слов в строке.
14. Удаляет все пробелы в строке.
15. Удаляет лишние пробелы между словами.
16. Удаляет все слова, в которых встречается заданный символ.
17. Заменяет все цифры в строке на символ «*».
18. Заменяет все пробелы в строке на дефисы.
19. Переворачивает каждое слово в строке.
20. Заменяет все заглавные буквы на строчные, а строчные на заглавные.
21. Удаляет все символы кроме букв, цифр и пробелов.
22. Дублирует каждое слово в строке.
23. Удаляет все слова, длина которых меньше 3 символов.

24. Удаляет все слова, длина которых больше 5 символов.
25. Добавляет скобки вокруг каждого слова.
26. Добавляет символ «#» перед каждым словом.
27. Переводит первую букву каждого слова в верхний регистр.

Вопросы к теоретическому материалу

1. Что называется строкой?
2. Укажите синтаксис объявления строк.
3. Каким образом можно задать длину строки?
4. Каким образом объявляются динамические строки?
5. В каких библиотеках содержатся функции для работы со строками?
6. Перечислите функции, применяемые для ввода строк.
7. Перечислите функции, применяемые для вывода строк.
8. Охарактеризуйте функцию gets.
9. Охарактеризуйте функцию puts.
10. Какая функция применяется для сравнения строк? Поясните ее работу.
11. Какая функция применяется для объединения строк? Поясните ее работу.
12. Какие функции применяются для копирования символов из одной строки в другую? Поясните их работу, укажите, в чем состоит их отличие.
13. Какая функция применяется для поиска подстроки в строке. Поясните ее работу

ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

По итогам выполнения каждой лабораторной работы студент:

1. Демонстрирует преподавателю правильно работающие программы;
2. Демонстрирует приобретенные теоретические знания, отвечая на пять вопросов по лабораторной работе;
3. Демонстрирует отчет по выполненной лабораторной работе, соответствующий всем требованиям.

Отчет по лабораторной работе оформляется по шаблону, представленному в приложении 1. Требования к отчету представлены в приложении 2.

ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №

по дисциплине

СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил

Студент гр.43__

Фамилия И.О.

Принял

Преподаватель Григорьева В.В.

Казань, 2025 г.

1. **Цель работы.**

2. **Задание на лабораторную работу** – вставляется задание на лабораторную работу, соответствующее индивидуальному, выданному преподавателем, варианту студента.

3. **Результат выполнения работы** – формируется описание хода выполнения работы и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны, например, *Рисунок 1. Начальное состояние программы* и т.п.).

4. **Листинг программы** – вставляется код разработанной программы

ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее 2 см.

Нумерация страниц – внизу страницы по центру, первая страница не нумеруется

Междустрочный интервал – 1,5 (полуторный), отступ первой строки – 1,25.

Текст документа должен быть выполнен с использованием шрифта Times New Roman, размер – 14, выравнивание – по ширине. Заголовки выполняются тем же шрифтом, но размера 16, полужирное начертание, размещение – по центру.

Рисунки должны размещаться по центру, они нумеруются по порядку. Перед рисунком в тексте на него должна быть ссылка. Подпись рисунка должна располагаться по центру и быть выполнена шрифтом Times New Roman, размер – 12. Сначала происходит нумерация рисунка, а затем пишется его название.