

Рабочая тетрадь по информатике

2 семестр

Группа _____

Студент _____ / _____ /

Дата сдачи _____

Преподаватель _____ / _____ /

Содержание

Содержание	2
Семинар 1. Первые программы на С	3
Семинар 2. Символьный ввод и вывод	5
Семинар 3. Массивы символов	7
Семинар 4. Типы данных, операции и выражения	9
Семинар 5. Управляющие конструкции и функции	10
Семинар 6. Функции. Рекурсия	11
Семинар 7. Указатели и массивы.	12
Семинар 8. Указатели и работа со строками.	14
Семинар 9. Форматированный ввод и вывод. Файлы.	15
Семинар 10. Структуры.	17

Семинар 1. Первые программы на С

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться компилировать и запускать программы на С.

Пример компиляции и запуска программы "Hello, World!":

```
> cc hello.c -o hello
> ./hello
Hello, World!
>
```

Чем можно пользоваться:

1. Функция printf
2. Циклы
3. Условия

Задание 1

Запустите программу "Hello, World!".

Задание 2

Выведите Ваше имя в рамке.

Пример:

```
*****
* Alexander *
*****
```

Задание 3

Вывести таблицу умножения чисел от 1 до 10. Пример вывода:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Контрольные вопросы

1. Какие особенности у функции `main`?
2. Что такое “управляющая последовательность”?
3. Какие основные возможности у функции `printf`?

Семинар 2. Символьный ввод и вывод

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться обрабатывать символьный ввод и вывод.

Чем можно пользоваться:

1. Функции `printf`, `getchar`, `putchar`
2. Циклы
3. Условия

Задание 1

Удалить из текста все пробелы и табуляции, а так же не печатать пустые строки. Пустая строка — строка, состоящая из одних пробелов или табуляций.

Пример:

```
> aaa bbb ccc
aaabbbccc
>
> aaa bbb
aaabbb
```

Задание 2

Вывести каждое слово с новой строки. Разделителями слов считаются пробел и табуляция, и они не считаются частью слова. Если пробелы или табуляции встречаются несколько раз подряд, то они должны быть пропущены при выводе.

Пример:

```
> aaa      bbb          ccc
aaa
bbb
ccc
```

Задание 3

Вывести только те символы, которые повторяются 3 или более раз подряд.

Пример:

```
> abc cab aaa def bbb eeee asdafeeeeeee
aaa
bbb
eeee
eeeeee
```

Контрольные вопросы

1. Что такое поток символов?
2. Чем отличаются символьная и строковая константы?
3. Что такое EOF?
4. Чему равна константа EOF?

Семинар 3. Массивы символов

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться работать со строками.

В языке C нет специального типа данных для строк. Строка в C — это массив символов.

Чем можно пользоваться:

1. Функции printf, getchar, putchar
2. Циклы
3. Условия

Задание 1

Напишите программу, которая бы разворачивала сокращенную запись наподобие a-z в строке s1 в полный список abc...xyz в строке s2, а так же запись z-a в zyx...cba. Программа должна работать с буквами в любом регистре и цифрами.

Пример:

```
> a-d
abcd
> e-a
edcba
> 0-9
0123456789
> a-c0-9
abc0123456789
> -a-c-f-
-abcdef-
```

Задание 2

Пользователь вводит в консоли 2 строки: s1 и s2. Напишите программу, удаляющую из строки s1 все символы, которые содержит строка s2. Удаление символов оформить в виде функции.

Пример:

```
> abcdef
> cde
abf
```

Задание 3

Пользователь вводит в консоли текст из буквы и цифр. Необходимо напечатать гистограмму частоты, с которой встречаются разные цифры в тексте. Необходимо разработать 2 варианта гистограммы: горизонтальную и вертикальную.

Пример горизонтальной гистограммы:

```
0: ####
1: ##
2: #####
...
9: ###
```

Пример вертикальной гистограммы:

```
      #      #
      #      #
#      #      #
# # # # # # #
0 1 2 3 4 5 6 7 8 9
```

Контрольные вопросы

1. Что такое массив?
2. Что такое '\0'?
3. Что такое кодировка и таблица кодировки?
4. В чем особенность передачи аргументов в функцию по значению?

Семинар 4. Типы данных, операции и выражения

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться работать со строками.

В языке C нет специального типа данных для строк. Строка в C — это массив символов.

Чем можно пользоваться:

1. Функции `printf`, `getchar`, `putchar`
2. Циклы
3. Условия

Задание 1

Пользователь вводит в консоли целое число в шестнадцатеричном формате. Напишите программу, которая преобразует строку в число. Перевод строки в число необходимо оформить в виде функции. Допустимые символы: 0–9, a–f, A–F.

Пример:

```
> FF
255
> 10
16
```

Задание 2

Напишите программу, которая будет посчитывать количество установленных нечетных бит в числе. При выполнении задания считаем, что биты нумеруются начиная с единицы. Для выполнения задания следует использовать побитовые операции.

Пример:

```
> FF
4
> 1
1
> 3
1
```

Контрольные вопросы

1. Преобразование типов?
2. Виды операций в C?
3. Как определяется приоритет операций?
4. В чем особенность передачи аргументов в функцию по значению?

Семинар 5. Управляющие конструкции и функции

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться декомпозировать программу на функции и компилировать программы, которые состоят из нескольких файлов.

Чем можно пользоваться:

1. Функции `printf`, `getchar`, `putchar`, библиотеки `ctype.h`, `math.h`.
2. Циклы
3. Условия

Задание 1

Пользователь вводит в консоли 2 строки: `s1` и `s2`. Напишите программу, которая находит положение первого встретившегося символа из `s1` в `s2`. Поиск следует оформить в виде функции. Если символ не найден, то следует выводить `-1`.

Пример:

```
> abcdef
> clkj
2
> abcdef
> lkj
-1
```

Задание 2

Модифицируйте программу калькулятор следующим образом:

1. Модифицируйте программу таким образом, чтобы функции `getch` и `ungetch` стали не нужны.
2. Добавьте поддержку операций: `%` целочисленное деление и `^` возведение в степень (функция `pow` из `math.h`)
3. Добавьте поддержку тригонометрических функций: `sin` и `cos` (см. `math.h`)
4. Дополнительное задание: расположите программу калькулятор в нескольких файлах. Напишите `Makefile` или `CMakeLists.txt` для компиляции программы при помощи утилиты `make` или `cmake` соответственно.

Пример:

```
> 2 3 ^ 3 %
2
```

Контрольные вопросы

1. Типы переменных?
2. Область видимости переменных?
3. Виды циклов и их особенности?
4. Заголовочные файлы?
5. Компиляция программы из нескольких файлов?

Семинар 6. Функции. Рекурсия

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться писать рекурсивные функции.

Задание 1

Реализуйте `swap` не как функцию, а как макрос, который будет применим к любому типу данных. Тип данных можно передавать в макрос как аргумент.

Задание 2

Перепишите алгоритм бинарного поиска, заменив цикл на рекурсию.

Задание 3

Перепишите алгоритм быстрой сортировки, заменив рекурсию на цикл.

Задание 4

Реализуйте как минимум любые 2 алгоритма сортировки из представленных ниже:

1. Поразрядная сортировка (Radix Sort)
2. Сортировка подсчетом (Counting Sort)
3. Сортировка слиянием (Merge Sort)
4. Один из вариантов улучшения алгоритма быстрой сортировки: randomized quick sort, 3-way quick sort, комбинированная быстрая сортировка (применение сортировки вставками на маленьких массивах), dual pivot quicksort
5. TimSort

Контрольные вопросы

1. Рекурсия?
2. Сортировка пузырьком? Преимущества и недостатки.
3. Сортировка Шелла? Преимущества и недостатки.
4. Быстрая сортировка? Преимущества и недостатки.
5. Когда стоит использовать макросы?

Семинар 7. Указатели и массивы.

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться писать программы, используя указатели.

Полезная информация:

Память для массива типа `T` длиной `N` элементов выделяется следующим образом:

```
T *a = (T*) malloc(N * sizeof(T));
```

Для целых чисел:

```
int *a = (int*) malloc(N * sizeof(int));
```

Освобождение памяти, которая была выделена при вызове функции `malloc`:

```
free(a);
```

Функции `malloc` и `free` находятся в библиотеке `stdlib.h`.

Считывание чисел из консоли при помощи функции `scanf`:

```
int i;
while (scanf("%d", &i) != EOF) {
    ...
};
```

В переменную `i` будут считаны все целые числа до конца ввода (достижения `EOF`).

Задание 1

Напишите программу для сортировки (сортировка Шелла) массива целых чисел произвольной длины. После запуска программы пользователь сначала должен указать длину массива, а затем перечислить все элементы массива.

Пример:

```
> 5
> 100
> 5
> 66
> 4
> 17
```

Результат:

```
> 4
> 5
> 17
> 66
> 100
```

Задание 2

Напишите функцию для считывания строки произвольной длины из консоли. Функция должна возвращать указатель на полученную строку. Строка должна завершаться символом конца строки.

Контрольные вопросы

1. Указатель?
2. Массив?
3. Разыменованье указателя?
4. Организация оперативной памяти?
5. Адресная арифметика?
6. Почему в `scanf` надо передавать указатель на переменную?

Семинар 8. Указатели и работа со строками.

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться писать программы, используя указатели, для обработки строк.

Задание 1

Напишите функцию `strend(char *s, char *t)`, которая возвращает 1, если строка `t` расположена в конце строки `s` и 0 в противном случае.

Задание 2

Напишите программу для сортировки массива строк (длину массива задает пользователь) согласно их длине. Строки должны занимать в памяти ровно столько места, сколько необходимо для их хранения. Для сортировки воспользуйтесь алгоритмом быстрой сортировки или сортировки Шелла.

Пример:

```
> 5
> abcd
> ab
> cde
> fdgdgfdfg
> fffff
```

Результат:

```
> ab
> cde
> abcd
> fffff
> fdgdgfdfg
```

Контрольные вопросы

1. Массив указателей?
2. Указатель на указатель?

Семинар 9. Форматированный ввод и вывод. Файлы.

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться работать с текстовыми файлами средствами стандартной библиотеки языка C.

Задание 1

Написать упрощенный аналог утилиты `tail`. Программа должна уметь выводить последние N строк файла. Имя (путь до файла) и N задаются аргументами командной строки. Значение N по умолчанию — 5. Пример:

Файл <code>file.txt</code>	Результат
abcd ab cde fdgdgfdg fffff abcd ab cde fdgdgfdg fffff	> ./tail file.txt abcd ab cde fdgdgfdg fffff > ./tail file.txt 3 cde fdgdgfdg fffff

Задание 2

Вывести строки из 2 файлов в перемешку: строка из одного файла, строка из другого файла. Файлы могут иметь разное количество строк. Когда один из файлов кончается следует вывести оставшееся содержимое второго. Имена файлов задаются аргументами командной строки. Пример:

Файл <code>file1.txt</code>	Файл <code>file2.txt</code>	Результат
abcd ab cde fdgdgfdg aaaa	cde dddddddd fffff	> ./prog file1.txt file2.txt abcd cde ab dddddddd cde fffff fdgdgfdg aaaa

Задание 3

Напишите программу, которая будет разбивать файл на страницы и выводить номер каждой строки. Строка на странице не может быть больше определенной длины. Количество строк на странице и максимальная длина строки на странице задаются через аргументы командной строки.

Файл file.txt	Результат
abcd ab cde fdgdgFdfg aaaa fdgd gfdfg aaaa	> ./page -p 2 -n 5 1: abcd 2: ab — Page 1 — 1: cde 2: fdgdg — Page 2 — 1: Fdfg 2: aaaa — Page 3 — 1: fdgd 2: gfdfg — Page 4 — 1: aaaa — Page 5 —

Контрольные вопросы

1. Текстовый поток?
2. Стандартный поток ввода?
3. Стандартный поток вывода?

Семинар 10. Структуры.

Дата	Оценка (макс. 2)	Подпись

Цель работы: научиться работать со структурами.

Задание 1

Написать программу, которая будет подсчитывать и выводить все уникальные слова из файла в алфавитном порядке и частоту их повторения. Работа программа должна быть основана на бинарном дереве. Имя файла передается как аргумент командной строки. Так же программа должна обрабатывать дополнительный необязательный аргумент командной строки: `-o ASC` и `-o DESC` — сортировка в порядке возрастания (значение по умолчанию) и возрастанию.

Пример:

Файл file.txt	Результат
abcd ab cde fdgdgfdg fffff cde ab cde	<pre>> ./unique file.txt ab 2 abcd 1 cde 3 fdgdgfdg 1 fffff 1 > ./unique -o DESC file.txt fffff 1 fdgdgfdg 1 cde 3 abcd 1 ab 2</pre>

Задание 2

Написать программу “база данных”. В базе должна храниться информация о городах и их населении. Имя города является уникальным ключом. Необходимые операции: добавить (`add`) и удалить (`remove`) информацию о городе, вывести весь список городов (`list`). Работа программа должна быть основана на хэш-таблице.

Пример:

<pre>> ./city_database Menu: 1 - add 2 - remove 3 - list 1 Moscow 15000000 1 London 10000000</pre>	<pre>1 Paris 5000000 3 Moscow 15000000 London 10000000 Paris 5000000 2 London</pre>	<pre>3 Moscow 15000000 Paris 5000000 1 Moscow 12000000 3 Moscow 12000000 Paris 5000000</pre>
---	---	--

Контрольные вопросы

1. Структура?
2. Структура с ссылкой на саму себя?
3. Бинарное дерево?
4. Хэш-таблица?