

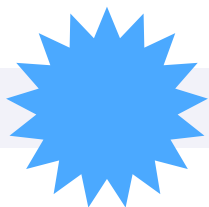
Программирование на C++



Минцифры
России

UCHi **DOMA**

20.35
УНИВЕРСИТЕТ

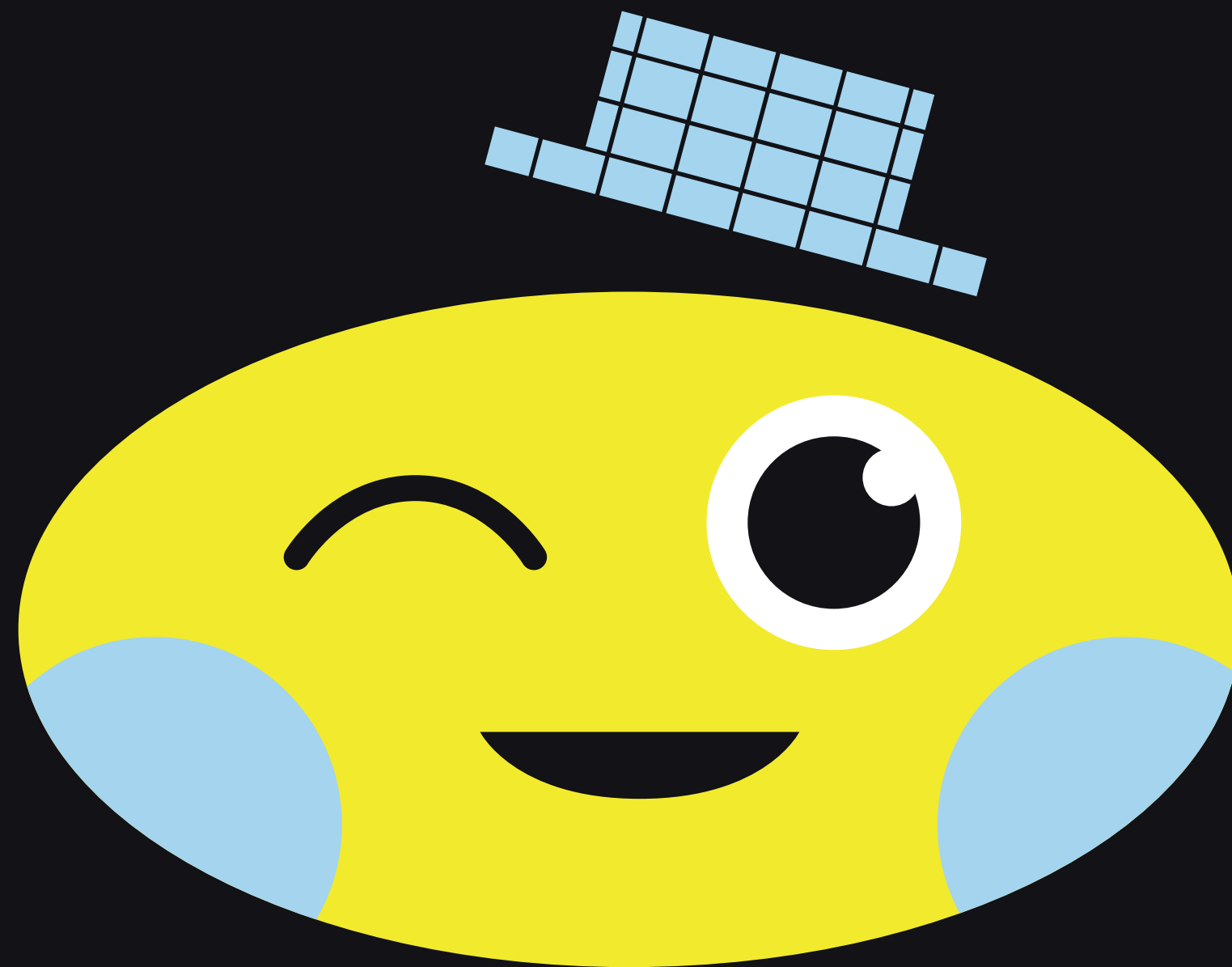


Модуль 2. Урок 4

Операторы работы с файлами



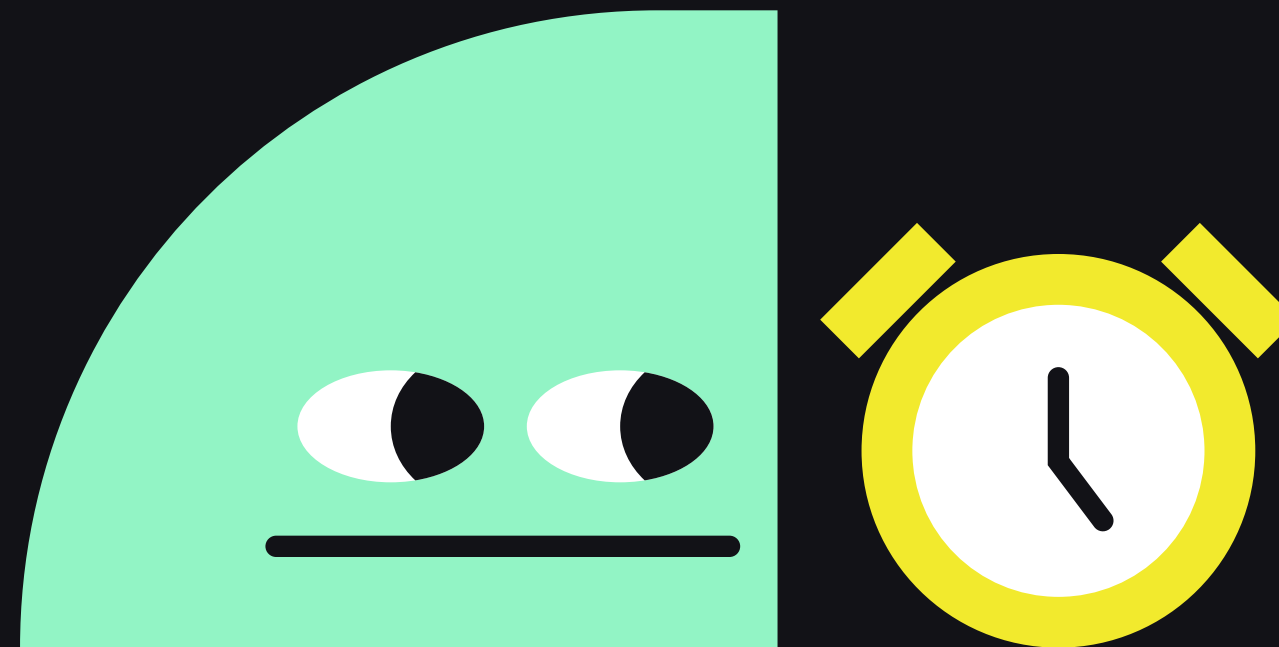
Привет!



проверка готовности



Видим и слышим друг друга без помех



Не опаздываем и не отвлекаемся



Сидим прямо



Улыбаемся, если всё ок

Как домашка?



Какие были трудности?



Какие остались вопросы?



Сколько заданий выполнено?



Разомнёмся



Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) + n, \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

```
1  #include <stdio.h>
2  int F(int n)
3  {
4      if (n==1) return 1;
5      else return F(n-1)+n;
6  }
7  int main()
8  {
9      int n;
10     scanf("%d", &n);
11     printf("%d\n", F(n));
12     return 0;
13 }
14
15
```

Разомнёмся



Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) + n, \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

Последовательно находим:

$$F(2) = F(1) + 2 = 3,$$

$$F(3) = F(2) + 3 = 6,$$

$$F(4) = F(3) + 4 = 10,$$

$$F(5) = F(4) + 5 = 15.$$

Задача



Написан алгоритм обработки данных, в результате получен большой массив данных, как их сохранить?



Написан алгоритм обработки данных, в результате получен большой массив данных, как его передать на другой компьютер?



Нужно обработать большой объем данных, вводить данные с клавиатуры долго и неудобно, как ввести данные для обработки?



Цели урока



изучить работу с файлами



отработать на практике
составление алгоритмов
считывающих данные
из файла и записывающих
информацию в файл на Си



Файл

Файл — именованный набор байтов, который может быть сохранен на некотором накопителе.

Данные, содержащиеся в файлах, имеют самый разнообразный характер:



программы



исходные данные для работы программ
или результаты выполнения программ



тексты



графические изображения



аудио



видео данные и т. п.

Каталог (папка, директория) — именованная совокупность байтов на носителе информации, содержащая название подкаталогов и файлов, используется в файловой системе для упрощения организации файлов.

Работа с файлами

- 1 Открыть файл, для того, чтобы к нему можно было обращаться. Соответственно, открывать можно для чтения, записи, чтения и записи, переписывания или записи в конец файла и т.п.
- 2 Непосредственно работа с файлом — запись и чтение. Работаем производится не с память с произвольным доступом, а с буферизированным потоком.
- 3 Заккрыть файл.

Работа с файлами

Для программиста открытый файл представляется как последовательность считываемых или записываемых данных. При открытии файла с ним связывается **поток ввода-вывода**. Выводимая информация записывается в поток, вводимая информация считывается из потока.

Когда поток открывается для ввода-вывода, он связывается со стандартной структурой типа **FILE**, которая определена в **stdio.h**. Структура **FILE** содержит необходимую информацию о файле.

Открытие файла

Для открытия файла используется функция **fopen()**, которая возвращает указатель на структуру типа **FILE**, который можно использовать для последующих операций с файлом.

```
FILE *fopen(name, type);
```

name

имя открываемого файла (включая путь)

type

указатель на строку символов, определяющих способ доступа к файлу:

"r"

открыть файл для чтения (файл должен существовать)

"w"

открыть пустой файл для записи; если файл существует, то его содержимое теряется

"a"

открыть файл для записи в конец (для добавления); файл создается, если он не существует

"r+"

открыть файл для чтения и записи (файл должен существовать)

"w+"

открыть пустой файл для чтения и записи; если файл существует, то его содержимое теряется

"a+"

открыть файл для чтения и дополнения, если файл не существует, то он создаётся

Возвращаемое значение — указатель на открытый поток. Если обнаружена ошибка, то возвращается значение NULL.

Заккрытие файла

Функция **fclose()** закрывает поток или потоки, связанные с открытыми при помощи функции **fopen()** файлами. Закрываемый поток определяется аргументом функции **fclose()**.

Возвращаемое значение



значение 0, если поток успешно закрыт



константа **EOF**, если произошла ошибка

Пример



```
1  #include <stdio.h>
2  int main()
3  {
4      FILE *fp;
5      char name[] = "name.txt";
6      if ((fp = fopen(name, "r")) == NULL)
7      {
8          printf("Не удалось открыть файл");
9          return 0;
10     }
11     // открыть файл удалось
12     ... // требуемые действия над данными
13     fclose(fp);
14     return 0;
15 }
```

Чтение символа из файла

```
char fgetc(поток);
```

Аргументом функции является указатель на поток типа **FILE**.
Функция возвращает код считанного символа. Если достигнут конец файла или возникла ошибка, возвращается константа **EOF**.

Чтение символа из файла

```
fgetc(символ, поток);
```

Аргументами функции являются символ и указатель на поток типа **FILE**.

Чтение строки из файла

```
fgets(УказательНаСтроку, КоличествоСимволов, поток);
```

Символы читаются из потока до тех пор, пока не будет прочитан символ новой строки ‘\n’, который включается в строку, или пока не наступит конец потока **EOF** или не будет прочитано максимальное количество символов. Результат помещается в указатель на строку и заканчивается нуль-символом ‘\0’.
Функция возвращает адрес строки.

Чтение строки из файла

```
fputs(УказательНаСтроку, поток);
```

Копирует строку в поток с текущей позиции.
Завершающий нуль-символ не копируется.

Запись в файл и чтение из файла

Функции `fscanf()` и `fprintf()` аналогичны функциям `scanf()` и `printf()`, но работают с файлами данных, и имеют первый аргумент — указатель на файл.

```
fscanf(поток, "ФорматВвода", аргументы);
```

```
fprintf(поток, "ФорматВывода", аргументы);
```

Пример



```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* S1, * S2;
7      int x, y;
8      system("chcp 1251");
9      system("cls");
10     printf("Введите число : ");
11     scanf("%d", &x);
12     S1 = fopen("S1.txt", "w");
13     fprintf(S1, "%d", x);
14     fclose(S1);
15     S1 = fopen("S1.txt", "r");
16     S2 = fopen("S2.txt", "w");
17     fscanf(S1, "%d", &y);
18     y *= 3;
19     fclose(S1);
20     fprintf(S2, "%d\n", y);
21     fclose(S2);
22     return 0;
23 }
```

S1 — Блокнот

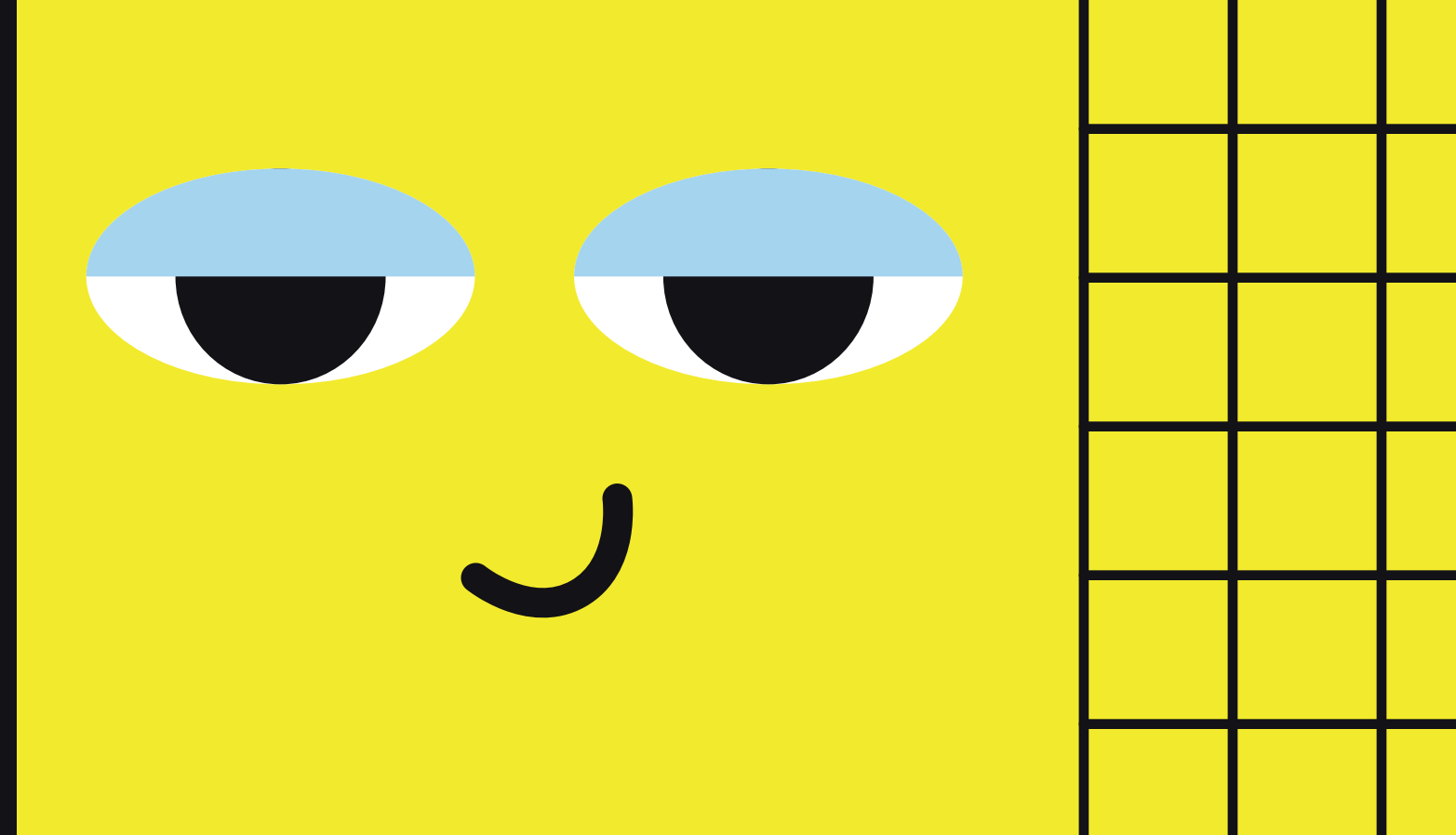
Файл Правка Формат Вид Справка

44

S2 — Блокнот

Файл Правка Формат Вид Справка

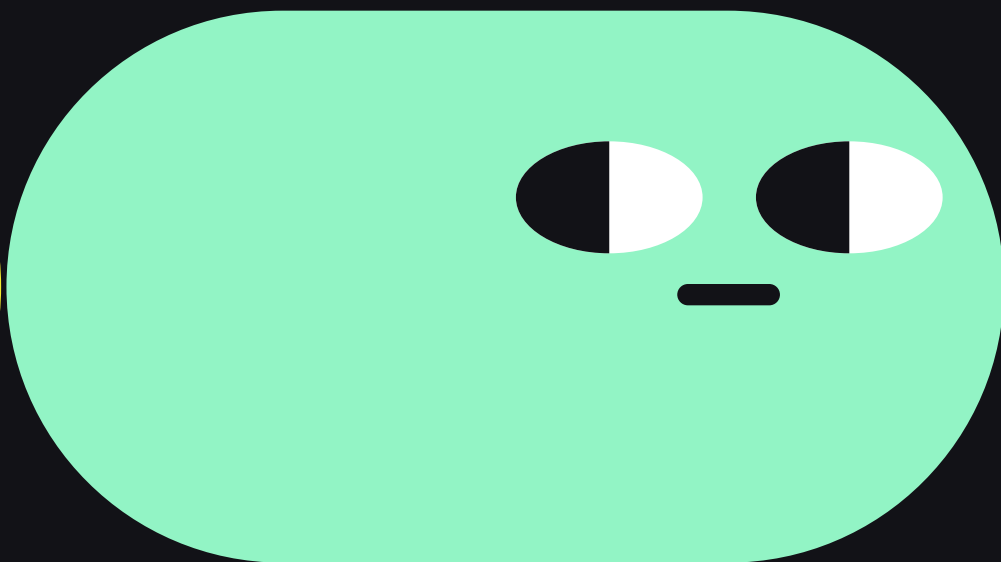
132



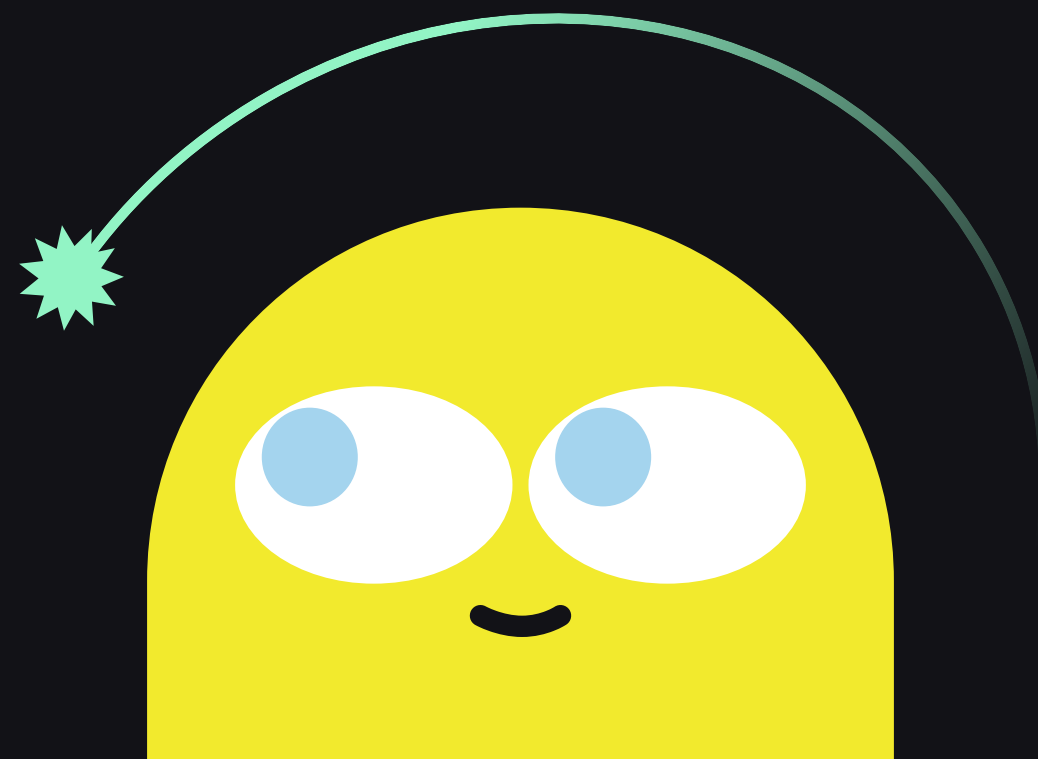
Практика

перерыв

физкультминутка



Смотрим вверх–вниз, вправо–влево



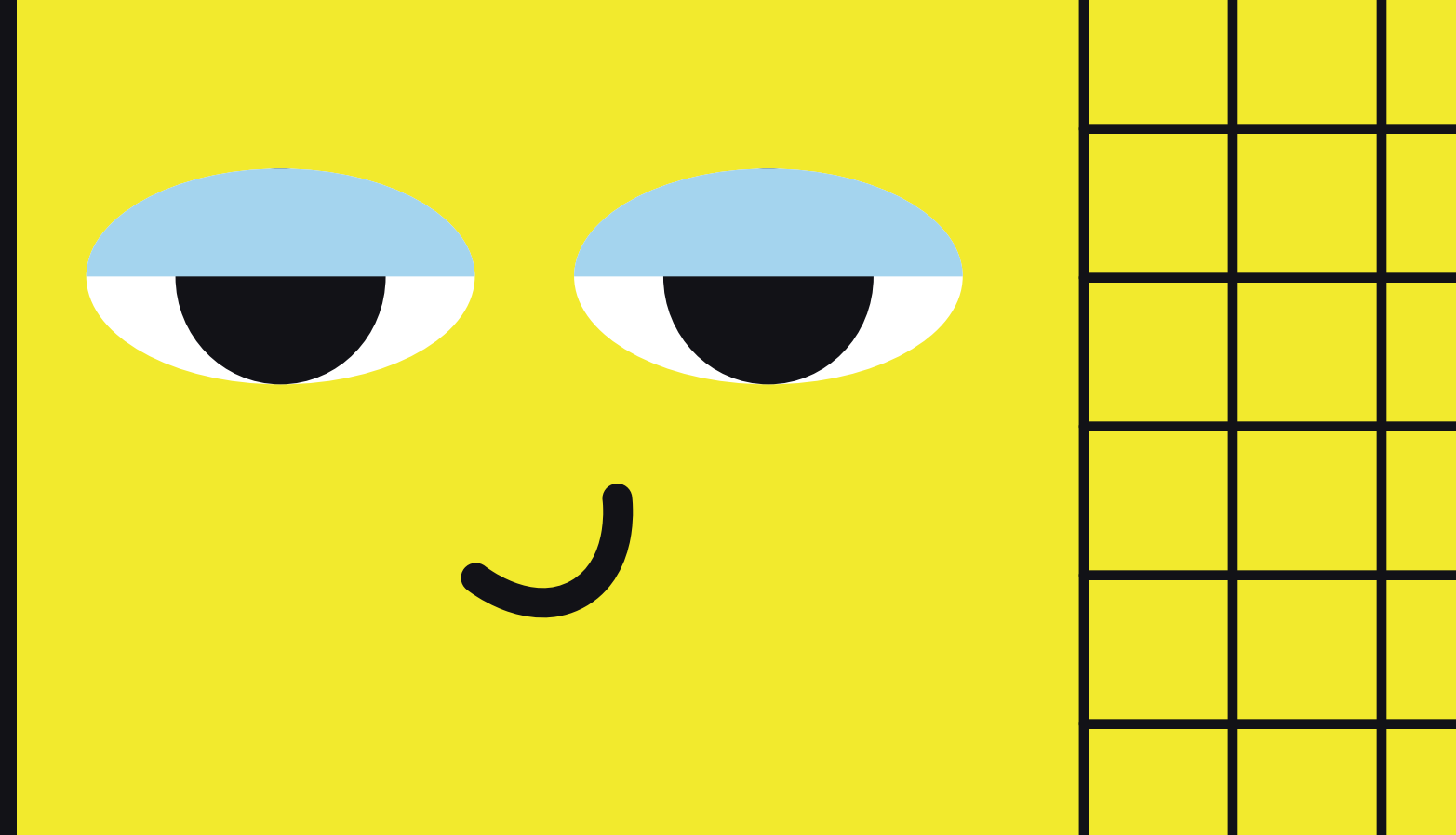
Вращаем по кругу туда–обратно



Крепко зажимаемся



Быстро моргаем



Практика

Задачи

1

Откройте файл «f1.txt», считайте данные из файла (целое число), возведите считанное значение во 2 степень и выведите на экран.

2

В файл записаны целые числа, по одному числу в строке. Считайте из файла «f1.txt» построчно все данные. Выведите на экран сумму считанных чисел.

3

В файл записаны целые числа, по одному числу в строке. Считайте из файла «f1.txt» построчно все данные. Удвойте значение каждого считанного значения и запишите в файл «f2.txt».

4

В файле «f1.txt» записан текст. Считать текст, определить сколько раз в считанном тексте встречается символ «s», вывести значение на экран.

5

Дан файл «f.txt». Создать два файла «f1.txt» и «f2.txt», записав в первый из них все четные числа, а во второй — все нечетные (каждое число в отдельной строке).

6

В файле «f.txt» записан текст. Перезаписать текст в файл «f1.txt» разделяя на строки, переходить на новую строку после каждого символа «s».

Задача 1

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input=NULL;
7      int y;
8
9      input = fopen("f1.txt", "r");
10     if (input == NULL)
11     {
12         printf("Error opening file");
13         exit(0);
14     }
15     fscanf(input, "%d", &y);
16     y *= y;
17     fclose(input);
18     printf("%d\n", y);
19     return 0;
20 }
```

Задача 2

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  ▼ int main() {
5      FILE* input=NULL;
6      int c;
7      int sum = 0;
8      input = fopen("f1.txt", "r");
9      if (input == NULL)
10 ▼ {
11     printf("Error opening file");
12     exit(0);
13 }
14 while (fscanf(input, "%d", &c) == 1)
15     sum += c;
16 fclose(input);
17 printf("%d", sum);
18 return 0;
19 }
```

Задача 3

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input=NULL;
7      FILE* output=NULL;
8      int c;
9
10     input = fopen("f1.txt", "r");
11     output = fopen("f2.txt", "w");
12     if (input == NULL)
13     {
14         printf("Error opening file f1.txt");
15         exit(0);
16     }
17     while (fscanf(input, "%d", &c) == 1)
18     {
19         c *= 2;
20         fprintf(output, "%dn\n", c);
21     }
22     fclose(input);
23     fclose(output);
24     return 0;
25 }
```

Задача 4

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input = NULL;
7      int c;
8      int count=0;
9
10     input = fopen("f1.txt", "r");
11     if (input == NULL)
12     {
13         printf("Error opening file f1.txt");
14         exit(0);
15     }
16     do
17     {
18         c = fgetc(input);
19         if (c == 's') count++;
20     } while (c != EOF);
21     fclose(input);
22     printf("%d", count);
23     return 0;
24 }
25
```

Задача 5

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input = NULL;
7      FILE* output1 = NULL;
8      FILE* output2 = NULL;
9      int c;
10     input = fopen("f.txt", "r");
11     output1 = fopen("f1.txt", "w");
12     output2 = fopen("f2.txt", "w");
13
14     if (input == NULL)
15     {
16         printf("Error opening file f1.txt");
17         exit(0);
18     }
19     while (fscanf(input, "%d", &c) == 1)
20         if (c%2==0) fprintf(output1, "%d\n", c);
21         else fprintf(output2, "%d\n", c);
22     fclose(input);
23     fclose(output1);
24     fclose(output2);
25     return 0;
26 }
```

Задача 6

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input = NULL;
7      FILE* output = NULL;
8      int c;
9      input = fopen("f1.txt", "r");
10     output = fopen("f2.txt", "w");
11     if (input == NULL)
12     {
13         printf("Error opening file f1.txt");
14         exit(0);
15     }
16     do
17     {
18         c = fgetc(input);
19         fprintf(output, "%c", c);
20         if (c == 's') fprintf(output, "\n");
21     } while (c != EOF);
22     fclose(input);
23     fclose(output);
24     return 0;
25 }
26
```

Закрепление

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input = NULL;
7      int c;
8      int sum = 0;
9
10     input = fopen("text.txt", "rt");
11     if (input == NULL) {
12         printf("Error opening file");
13         exit(0);
14     }
15     while (fscanf(input, "%d", &c) == 1)
16         sum += c;
17     fclose(input);
18     printf("%d", sum);
19     return 0;
20 }
```

Что будет выведено на экран в результате работы экрана?

text — Блокнот

Файл Правка Формат Вид Справка

1

2

3

Закрепление

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main()
5  {
6      FILE* input = NULL;
7      int c;
8      int sum = 0;
9
10     input = fopen("text.txt", "rt");
11     if (input == NULL) {
12         printf("Error opening file");
13         exit(0);
14     }
15     while (fscanf(input, "%d", &c) == 1)
16         sum += c;
17     fclose(input);
18     printf("%d", sum);
19     return 0;
20 }
```

Что будет выведено на экран в результате работы экрана?

Результат работы программы

6

text — Блокнот

Файл Правка Формат Вид Справка

1

2

3



Подведём итоги



изучили работу с файлами



отработали на практике
составление алгоритмов
считывающих данные
из файла и записывающих
информацию в файл на Си

Оцени сложность урока

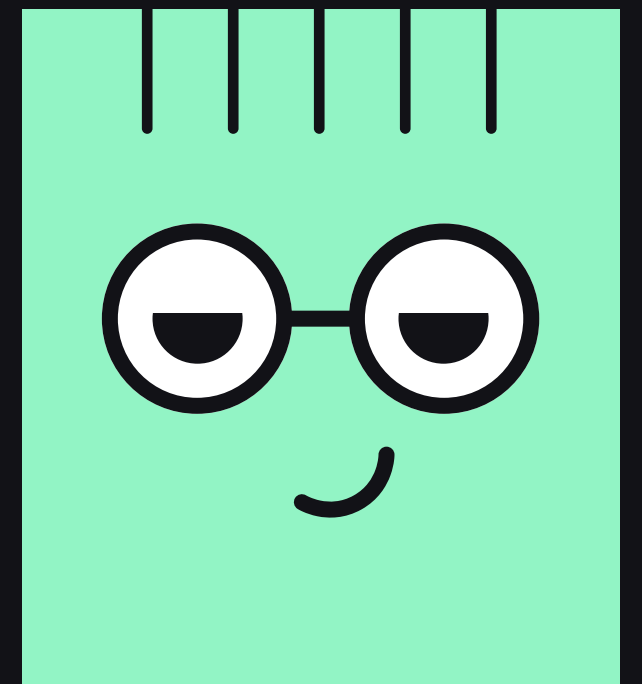
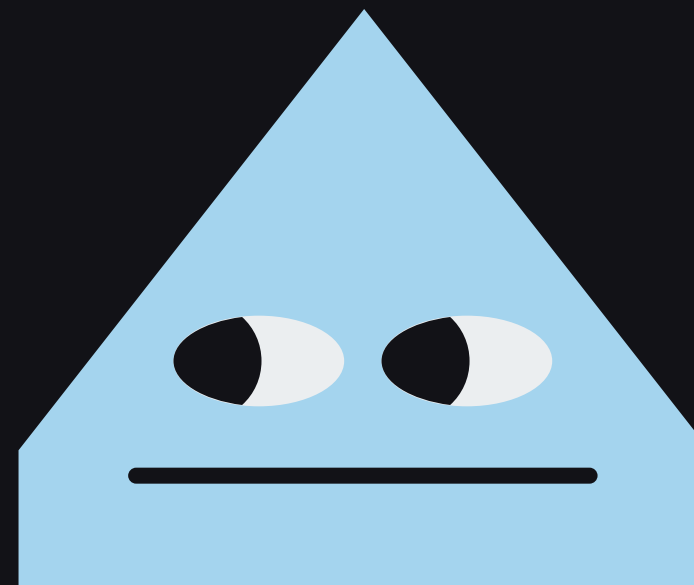
1 если тебе было совсем просто

2 было достаточно просто, но ты узнал(а) что-то новое

3 было не очень просто, но достаточно комфортно, ты узнал(а) много нового

4 было сложно, ты не знал(а) ничего из материала

5 было слишком сложно, многое осталось для тебя непонятным



Домашнее задание



До встречи!