

Занятие 4

Циклы for, while, do while. Операторы ветвления if и switch, логические операции. Библиотека cctype. Простые подпрограммы.

Пример 1. Цикл for.

```
1 #include <iostream>
2 int main(void){
3     using namespace std;
4
5     int a[] = {10, 11, 12, 13};    // объявление и инициализация массива
6     int N = sizeof(a)/sizeof(a[0]); // находим количество элементов
7
8     for (int i = 0; i < N; i++){
9         cout << "a[" << i << "] = " << a[i] << endl; // вывод элементов массива
10    }
11    return 0;
12 }
```

Пример 2. Цикл while.

```
1 #include <iostream>
2 int main(void){
3     using namespace std;
4     const int ArSize = 30;
5     char name[ArSize];
6     cout << "Your first name, please: ";
7     cin >> name; // ввод имени
8
9     // Вывод имени посимвольно и в кодах ASCII:
10    cout << "Here is your name, verticalized and ASCIIized:\n";
11    int i = 0; // начать с начала строки
12    while (name[i] != '\0'){ // обрабатывать до конца строки
13        cout << name[i] << " : " << int(name[i]) << endl;
14        i++; // не забудьте этот шаг
15    }
16    return 0;
17 }
```

Пример 3. Цикл do-while с проверкой на выходе.

```
1 #include <iostream>
2 int main(void){
3     using namespace std;
4     int n;
5
```

```

6 // Запрос на ввод любимого числа из диапазона 1-10
7 cout << "Enter numbers in the range 1-10 to find ";
8 cout << "my favorite number" << endl;
9
10 do{
11     cin >> n; // выполнить тело
12 } while (n != 7); // затем проверить
13
14 cout << "Yes, 7 is my favorite. \n"; // 7 - любимое число
15
16 return 0;
17 }

```

Пример 4. Использование оператора if else. Библиотека cctype (ctype.h)

Заголовочный файл cctype (или ctype.h) содержит декларации функций, которые определяют тип символов и могут их преобразовывать. Например, функция int isalpha(int c) возвращает ненулевое значение (true), если её аргумент является буквой, в противном случае возвращается нуль (false). Пример обработки текста с помощью различных функций приведён ниже.

```

1 #include <iostream>
2 #include <cctype> // прототипы символьных функций
3 int main(void){
4     using namespace std;
5     // запрос текста для анализа; завершающий символ - @
6     cout << "Enter text for analysis, and type @ to terminate input. \n";
7
8     char ch;
9     int whitespace = 0, digits = 0, chars = 0, punct = 0, others = 0;
10    cin.get(ch); // получение первого символа
11    while (ch != '@'){ // проверка на признак окончания ввода
12        if (isalpha(ch)) // буквенный символ?
13            chars++;
14        else if (isspace(ch)) // пробельный символ?
15            whitespace++;
16        else if (isdigit(ch)) // десятичная цифра?
17            digits++;
18        else if (ispunct(ch)) // знак препинания?
19            punct++;
20        else
21            others++;
22        cin.get(ch); // получение следующего символа
23    }
24    // вывод на экран:
25    cout << chars << " letters, " // количество букв,
26         << whitespace << " whitespace, " // пробелов,
27         << digits << " digits, " // цифр,
28         << punct << " punctuations, " // знаков препинания
29         << others << " others.\n"; // и прочих символов
30    return 0;
31 }

```

Пример 5. Использование оператора switch. Простые подпрограммы.

Разбиение программы на составные части (подпрограммы или функции) позволяет значительно упростить разработку программы, сделать её более понятной и сократить коли-

чество ошибок за счет использования уже существующих функций.

Для использования какой-либо функции должно быть дано её определение. Определение включает имя функции, тип аргументов, тип возвращаемого значения и непосредственно последовательность операций, осуществляемых функцией (тело функции). Например, определение функции `show_menu()` дано в строках 4-7 приведенного ниже кода. Помимо определения существует также такое понятие как прототип (описание, декларация) функции, которое включает в себя имя функции, типы аргументов и возвращаемого значения, но без тела функции. Например, декларация (прототип) функции `show_menu()` выглядит так:

```
void show_menu(void);
```

При компилировании кода, в котором используется какая-либо функция, компилятор должен знать прототип этой функции, то есть имя функции, тип аргументов и тип возвращаемого значения. Это достигается двумя способами. Первый способ – это поместить определение функции перед её вызовом в том же файле. Второй способ – это поместить определение функции в отдельном файле, а в файле, где эта функция используется, поместить декларацию (прототип) функции перед её вызовом (обычно в самом начале файла). При этом декларация может быть прописана как в явном виде, так и путём включения с помощью команды `#include` заголовочного файла с прототипом этой функции. Также можно давать определение функции после её вызова в том же файле, но тогда надо в явном виде давать прототип перед вызовом.

В примере ниже все определения отдельных функций (`show_menu`, `to_8`, `to_16`) даны в начале файла перед их вызовом.

```
1 #include <iostream>
2 using namespace std;
3
4 void show_menu(void) {
5     cout << "Please enter 1, 2, or 3:\n"
6         << "1) to base-8 \n2) to base-16 \n3) quit \n";
7 }
8 void to_8(int n) {
9     cout << "base-8: " << oct << n << endl;
10 }
11 void to_16(int n) {
12     cout << "base-16: " << hex << n << endl;
13 }
14
15 int main(void){
16     cout << "Program to transform a base-10 integer " <<
17         << "to base-8 or base-16 representation" << endl;
18
19     int n;
20     cout << "Enter an integer (base-10) -> ";
21     cin >> n;
22     show_menu();
23     int choice;
24     cin >> choice;
25     while (choice != 3) {
26         switch (choice) {
27             case 1:
28                 to_8(n);
```

```

29     break;
30     case 2:
31         to_16(n);
32         break;
33     default:
34         cout << "That's not a choice. \n";
35     }
36     show_menu();
37     cin >> choice;
38 }
39 cout << "Bye!\n";
40 return 0;
41 }

```

Задания

Задание 4.1. Напишите программу, которая использует объект `string` (или массив `char`) и цикл для чтения по одному слову за раз до тех пор, пока не будет введено слово "done". Затем программа должна сообщить количество введенных слов (исключая "done").

Задание 4.2. Напишите программу, использующую вложенные циклы, которая запрашивает у пользователя значение количества строк для отображения. Затем она должна отобразить указанное число строк со звездочками: с одной звездочкой в первой строке, двумя – во второй и т.д. В каждой строке звездочкам должны предшествовать знаки `#` в таком количестве, чтобы общее число символов в каждой строке было равно количеству строк. Пример запуска программы должен выглядеть следующим образом:

Введите количество строк: 5

```

####*
####*
####*
####*
*****

```

Задание 4.3. Напишите программу, запрашивающую у пользователя ввод двух целых чисел. Затем программа должна вычислить (используя цикл) и вывести на экран сумму всех целых чисел, лежащих между этими двумя числами, включая их. Предполагается, что меньшее значение вводится первым.

Задание 4.4. Напишите программу, которая читает слова по одному за раз, пока не будет введена отдельная буква `q`. После этого программа должна сообщить количество слов, начинающихся с гласных, количество слов, начинающихся с согласных, а также количество слов, не попадающих в эти категории. Одним из возможных подходов может быть применение `isalpha()` для различения слов, начинающихся с букв, с последующим применением `if` или `switch` для идентификации слов, начинающихся, например, с гласных. Запуск программы может выглядеть так:

```

Enter words (q to quit):
The 12 awesome oxen ambled

```

quietly across 15 meters of lawn, q

5 words beginning with vowels

4 words beginning with consonants

2 others

Задание 4.5. Предположим, что Вы продаете книгу по программированию на C++. Напишите программу, которая позволяет вести ежемесячный учет количества проданных книг. Программа должна использовать цикл, в котором выводится запрос на ввод данных с названием месяца, применяя массив объектов `string` (или массив указателей на `char`), инициализированный названиями месяцев. Введенное количество проданных книг сохраняется в массиве типа `int`. Затем программа должна найти общее количество проданных книг и вывести его на экран.

Задание 4.6. В США используют прогрессивную шкалу федерального налогообложения, которая меняется ежегодно. В 2019 году для индивидуальных налогоплательщиков она выглядела следующим образом: из годового дохода вычитается \$12 200 (необлагаемая налогом часть), а затем оставшаяся часть облагается налогом по следующим ставкам: на первые \$9 700 – 10%, на следующие от \$9 701 до \$39 475 – 12%, на следующие от \$39 476 до \$84 200 – 22%, на следующие от \$84 201 до \$204 100 – 32%, на следующие от \$204 101 до \$510 300 – 35%, на следующие от \$510 301 и выше – 37%.

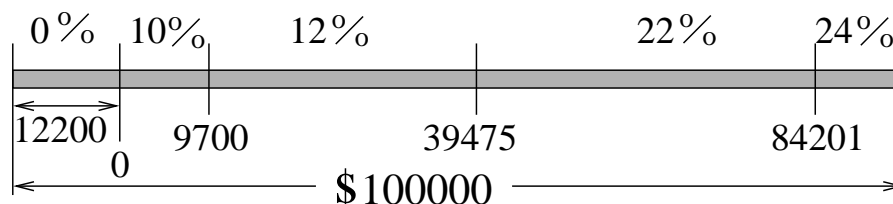


Рис. 1: Распределение ставки налога на примере годового дохода в \$100 000.

Напишите программу, которая использует цикл для запроса годового дохода и расчета подлежащего к уплате налога. Также программа должна рассчитать эффективную ставку налога. Эффективная ставка налога – это такая ставка, которая соответствует уплаченному налогу, если бы вся сумма дохода облагалась налогом по единой ставке. Цикл должен завершаться, когда пользователь вводит отрицательное или нечисловое значение. Согласно данным Бюро по статистике трудовой деятельности (Bureau of Labor Statistics) медианная годовая зарплата в США на конец 2019 года составляла \$48 672. Какой будет налог и эффективная ставка при такой зарплате?