

# Программирование

И. Д. Липанов

24 января 2016 г.

# Оглавление

<b>1</b>	<b>Основные конструкции языка</b>	<b>3</b>
1.1	Задание 2 . . . . .	3
1.1.1	Задание . . . . .	3
1.1.2	Теоретические сведения . . . . .	3
1.1.3	Проектирование . . . . .	4
1.1.4	Описание тестового стенда и методики тестирования	4
1.1.5	Выводы . . . . .	4
<b>2</b>	<b>Циклы</b>	<b>5</b>
2.1	Задание 1 . . . . .	5
2.1.1	Задание . . . . .	5
2.1.2	Теоретические сведения . . . . .	5
2.1.3	Проектирование . . . . .	5
2.1.4	Описание тестового стенда и методики тестирования	6
2.1.5	Тестовый план и результаты тестирования . . . . .	6
2.1.6	Выводы . . . . .	6
<b>3</b>	<b>Массивы</b>	<b>7</b>
3.1	Задание 2 . . . . .	7
3.1.1	Задание . . . . .	7
3.1.2	Теоретические сведения . . . . .	7
3.1.3	Проектирование . . . . .	7
3.1.4	Описание тестового стенда и методики тестирования	8
3.1.5	Выводы . . . . .	8
<b>4</b>	<b>Арифметика</b>	<b>10</b>
4.1	Задание 1 . . . . .	10
4.1.1	Задание . . . . .	10
4.1.2	Теоретические сведения . . . . .	10
4.1.3	Проектирование . . . . .	10
4.1.4	Описание тестового стенда и методики тестирования	11

4.1.5	Выводы . . . . .	11
<b>5</b>	<b>Задание на строки</b>	<b>12</b>
5.1	Табличная функция . . . . .	12
5.1.1	Задание . . . . .	12
5.1.2	Теоретические сведения . . . . .	12
5.1.3	Проектирование . . . . .	12
5.1.4	Описание тестового стенда и методики тестирования	13
5.1.5	Тестовый план и результаты тестирования . . . . .	13
5.1.6	Выводы . . . . .	13

# Глава 1

## Основные конструкции языка

### 1.1 Задание 2

#### 1.1.1 Задание

Пользователь задает сумму денег в рублях, меньшую 100 (например, 16). Определить, как выдать эту сумму монетами по 5, 2 и 1 рубль, израсходовав наименьшее количество монет (например,  $3 \times 5p + 0 \times 2p + 1 \times 1p$ ).

#### 1.1.2 Теоретические сведения

Было использовано:

- функции для ввода и вывода информации, образцы которых находятся в `<stdio.h>`
- функция считывания символа из консоли без отображения, прототип которой находится в `<conio.h>`
- использовался цикл "for" для создания циклов, которые должны выполняться заданное число раз

Для решения задачи требовалось знать некоторые стандартные функции языка C.

Необходимое наименьшее кол-во монет было найдено с помощью функций ввода-вывода. Также несоизмеримо помог тип данных "int" с помощью которого мы объявляли номиналы монет и их кол-во.

$$const int n = 3; const int c[n] = 5, 2, 1; \tag{1.1}$$

### 1.1.3 Проектирование

Использовалась только одна функция для взаимодействия с пользователем (так как под каждую задачу я выделял отдельный проект)

- Реализация задачи была решена функцией `main.cpp`.

### 1.1.4 Описание тестового стенда и методики тестирования

Использовался Qt Creator 3.5.0 (opensource) с GCC 4.9.1 компилятором  
Операционная система: Windows 10

Ручное тестирование почти отсутствовало. Для статического анализа был использован `sppcheckgui` версии 1.7.1 . Ошибок и предупреждений не было. Для автоматического тестирования был использован `framework qt test`, с помощью которого были реализованы модульные тесты.

### 1.1.5 Выводы

При написании данной программы никаких трудностей не было. Никаких сложных арифметических вычислений для написания программы не требовалось.

### Листинги

```
1 #include <math.h>
2 #include <stdio.h>
3 #include <conio.h>
4 const int n=3;
5 const int c[n]={5,2,1};
6 int main()
7 {
8     int coins[n];
9     int money;
10    printf("sum of money: ");
11    scanf("%d",&money);
12    for (int i=0;i<n;i++)
13    {
14        coins[i]= money / c[i];
15        printf("\n%d rub. - %d coin",c[i],coins[i]);
16        money=money % c[i];
17    }
18    getch();
19    return 0;
20 }
```

# Глава 2

## ЦИКЛЫ

### 2.1 Задание 1

#### 2.1.1 Задание

Текст содержит следующие знаки корректуры: \$ - сделать красную строку, # - удалить следующее слово, @ - удалить следующее предложение. Произвести указанную корректировку.

#### 2.1.2 Теоретические сведения

Было использовано:

- функции для ввода и вывода из файла
- конструкция "if"
- конструкция "while"

Для решения данной задачи необходимо было уметь считывать и записывать информацию в файл, а также уметь корректировать текст.

Требовалось считывать текст из файла, делать проверку на наличие определенных символов и записывать исправленный текст в файл.

#### 2.1.3 Проектирование

Было решено выделить одну функцию:

- `strings.c` для нахождения символов и проведения корректировки текста.

### 2.1.4 Описание тестового стенда и методики тестирования

Использовался QtCreator с GCC компилятором. Операционная система: Windows 10

Использовалось ручное тестирование, автоматическое тестирование не проводилось. Ошибок и предупреждений не возникало.

### 2.1.5 Тестовый план и результаты тестирования

Все тесты были пройдены успешно: полученный результат, совпал с ожидаемым.

### 2.1.6 Выводы

В ходе написания программы не возникло никаких трудностей

### Листинги

```
1  #include <cstring>
2  #include <iostream>
3  #include <stdio.h>
4
5  using namespace std;
6
7  void read_word(char *s, int *i, char *w) //чтение слова
8  {
9      int j;
10     while (s[*i] <= ' ')
11         (*i)++;
12     j = 0;
13     while (s[*i] != '\0' && s[*i] != ' ')
14     {
15         w[j] = s[*i];
16         j++;
17         (*i)++;
18     }
19     w[j] = '\0';
20 }
21 void strings()
22 {
23     FILE *f,*g;
24     char s[2000], w[2000], ws[2000], pred[2000], pred1[2000];
25     s[0] = '\0';
26     ws[0] = '\0';
```

```

27     w[0] = '\0';
28     int vsp=1;
29     int sl=1;
30     int str;
31     int j = 0;
32     f = fopen("2.txt", "r");
33     if (!f)
34     {
35         puts("cannot open the file");
36         return;
37     }
38     while (!feof(f))
39     {
40         fgets(s, 2000, f);
41
42         int i = 0;
43         str = 0;
44         ws[0] = '\0';
45         if (vsp)
46             while (s[i] != '\0')
47             {
48                 if (s[i] == '$')
49                 {
50                     w[j] = '\n';
51                     j++;
52                     w[j] = ' ';
53                     j++;
54                     w[j] = ' ';
55                     j++;
56                     w[j] = '\0';
57                     i++;
58                 }
59                 else
60                 {
61                     if (s[i] == '#')
62                         read_word(s, &i, ws);
63                     else
64                     {
65                         if (s[i] == '@')
66                         {
67                             while (s[i + 1] != '.' && s[i + 1] != '
! ' && s[i + 1] != '?' && s[i + 1] != '
\0')
68                                 i++;
69                             i++;
70                             if (s[i] == '\0')
71                                 vsp = 0;
72                         }
73                     }

```



```

74         {
75             w[j] = s[i];
76             i++;
77             j++;
78             w[j] = '\0';
79         }
80     }
81 }
82 }
83 else
84 {
85     while (s[i + 1] != '.' && s[i + 1] != '!' && s[i + 1]
86            != '?' && s[i + 1] != '\0')
87         i++;
88     if (s[i] != '\0')
89         vsp = 1;
90 }
91
92 }
93 printf("text:\n");
94 puts(w);
95 fclose(f);
96 remove("2.txt");
97 g = fopen("2.txt", "w");
98 fputs(w, g);
99 fclose(g);
100 getchar();
101 getchar();
102 }

```

# Глава 3

## Массивы

### 3.1 Задание 2

#### 3.1.1 Задание

На шахматной доске стоят три ферзя (ферзь бьет по вертикали, горизонтали и диагоналям). Найти те пары из них, которые угрожают друг другу. Координаты ферзей вводить целыми числами.

#### 3.1.2 Теоретические сведения

Было использовано:

- несколько функций для ввода и вывода информации, образцы которых находятся в `<stdio.h>`
- оператор "if" обеспечивающий выборочное выполнение отдельных участков кода

Потребовались знания синтаксиса языка C.

#### 3.1.3 Проектирование

3 раза была использована адресная арифметика:

- $((x1==x2) || (y1==y2) || (abs(x1-x2)==abs(y1-y2)))$
- $((x1==x3) || (y1==y3) || (abs(x1-x3)==abs(y1-y3)))$
- $((x2==x3) || (y2==y3) || (abs(x2-x3)==abs(y2-y3)))$

### 3.1.4 Описание тестового стенда и методики тестирования

Использовался QtCreator с GCC компилятором. Операционная система: Windows 10

Ручное тестирование присутствовало. Для статического анализа был использован `cppcheckgui`. Ошибок и предупреждений не было. Для автоматического тестирования был использован `framework qt test`, с помощью которого были реализованы модульные тесты.

### 3.1.5 Выводы

Проблемы были в адресной арифметике, но после некоторых изменений в местоположениях указателей результат был схож с предполагаемым.

#### Листинги

```
1 #include <stdio.h>
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int x1,x2,x3,y1,y2,y3,k;
8     scanf("%d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3 );
9     ;
10    k=0;
11    if ((x1==x2)|| (y1==y2)|| (abs(x1-x2)==abs(y1-y2)))
12    {
13        printf("The threat between 1 and 2 a pair of Queens\n"
14              );
15        k++;
16    }
17    if ((x1==x3)|| (y1==y3)|| (abs(x1-x3)==abs(y1-y3)))
18    {
19        printf("The threat between 1 and 3 a pair of Queens\n"
20              );
21        k++;
22    }
23    if ((x2==x3)|| (y2==y3)|| (abs(x2-x3)==abs(y2-y3)))
24    {
25        printf("The threat between 2 and 3 a pair of Queens\n"
26              );
27        k++;
28    }
29 }
```

```
26     if (k==0) printf("There's no threat\n");
27
28     return 0;
29 }
30
31 // [4 6 1 3 4 2] - "The threat between 1 and 2 a pair of
   Queens"
32 // [2 9 3 6 6 3] - "The threat between 2 and 3 a pair of
   Queens"
33 // [4 4 5 6 7 4] - "The threat between 1 and 3 a pair of
   Queens"
34 // [2 9 3 7 4 4] - "There's no threat"
```

## Глава 4

# Арифметика

### 4.1 Задание 1

#### 4.1.1 Задание

Найти корни квадратного уравнения:  $y=ax^2+bx+c$ .

#### 4.1.2 Теоретические сведения

Было использовано:

- функции для ввода-вывода информации, образцы которых находятся в `<stdio.h>`
- функция вычисления квадратного корня, образец которой находится в `<math.h>`

Для решения поставленной задачи требовалось знание основ синтаксиса языка C и умение решать квадратные уравнения.

#### 4.1.3 Проектирование

Была использована только одна функция `main.cpp` так как программа была выполнена в отдельном проекте.

- `remove_characters_from_the_string` в ней реализовано удаление ненужных символов, и замена символов больших регистров на маленькие.
- `find_the_longest_substring` в этой функции реализован алгоритм поиска самой длинной подстроки

#### 4.1.4 Описание тестового стенда и методики тестирования

Использовался QtCreator с GCC компилятором. Операционная система: Windows 10

Ручное тестирование присутствовало. Для статического анализа был использован `sppcheckgui`. Ошибок и предупреждений не было. Для автоматического тестирования был использован `framework qt test`, с помощью которого были реализованы модульные тесты.

#### 4.1.5 Выводы

При написании программы не возникло особых проблем.

#### Листинги

```
1 #include <iostream>
2 #include <math.h>
3 #include <stdio.h>
4 using namespace std;
5
6 int main()
7 {
8     double a,b,c,d,e,f;
9     cout<<"Enter the factor a,b,c"<<endl;
10    cin>>a>>b>>c;
11    d=(b*b)-(4*a*c);
12
13    if(d==0)
14    {
15        e=(-b+sqrt(d))/(2*a);
16        cout<<"x ="<<e<<endl;
17    }
18    if(d>0)
19    {
20        e=(-b+sqrt(d))/(2*a);
21        f=(-b-sqrt(d))/(2*a);
22        cout<<"e ="<<e<<endl;
23        cout<<"f ="<<f<<endl;
24    }
25    if(d<0)
26    {
27        cout<<"No roots"<<endl;
28    }
29 }
```

## Глава 5

# Задание на строки

### 5.1 Табличная функция

#### 5.1.1 Задание

Текст содержит многократно вложенные круглые скобки. Исправить его, оставив скобки первого уровня круглыми, второго – заменить на квадратные, третьего и последующих – на фигурные.

#### 5.1.2 Теоретические сведения

Было использовано:

- `<iostream>` и пространство имён `std`, для взаимодействия с пользователем через консоль.
- `<cstring>` класс с методами и переменными для организации работы со строками

Умение считывать и записывать информацию в файл.

#### 5.1.3 Проектирование

Была выделена одна функция:

- `cstrings.c` для нахождения символов и проведения корректировки скобок.

### 5.1.4 Описание тестового стенда и методики тестирования

Использовался QtCreator с GCC компилятором. Операционная система: Windows 10

Ручное тестирование почти отсутствовало. Для автоматического тестирования был использован framework qt test, с помощью которого были реализованы модульные тесты.

### 5.1.5 Тестовый план и результаты тестирования

Была произведена корректировка скобок. Все тесты были пройдены успешно: полученный результат, совпал с ожидаемым.

### 5.1.6 Выводы

В ходе написания программы не возникло никаких трудностей.

## Листинги

```
1 #include <cstring>
2 #include <iostream>
3
4 using namespace std;
5
6 int main()
7 {
8     char s[100];
9     cin >> s;
10    int l = 0;
11    int maxl = 0;
12    for( int i=0 ; i < strlen(s); i++ )
13    {
14        if( s[i] == '(' )
15        {
16            s[i] = '{';
17            l++;
18            if( l > maxl )
19            {
20                maxl = l;
21            }
22        }
23        if( s[i] == ')' )
24        {
25            s[i] = '}';
```



```

26         l--;
27     }
28 }
29 for( int i=0 ; i < strlen(s); i++ )
30 {
31
32     if( s[i] == '{' )
33     {
34         l++;
35         if( l == maxl ) s[i] = '(';
36         if( l == maxl-1 ) s[i] = '[';
37     }
38     if( s[i] == '}' )
39     {
40         if( l == maxl ) s[i] = ')';
41         if( l == maxl-1 ) s[i] = ']';
42         l--;
43     }
44 }
45 cout << s << endl;
46 }

```