

Программирование

И. Д. Липанов

27 января 2016 г.

Оглавление

1	Основные конструкции языка	3
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
2	Циклы	5
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
3	Массивы	9
3.1	Задание 2	9
3.1.1	Задание	9
3.1.2	Теоретические сведения	9
3.1.3	Проектирование	9
3.1.4	Описание тестового стенда и методики тестирования	10
3.1.5	Выводы	10
4	Арифметика	12
4.1	Задание 1	12
4.1.1	Задание	12
4.1.2	Теоретические сведения	12
4.1.3	Проектирование	12
4.1.4	Описание тестового стенда и методики тестирования	13

4.1.5	Выводы	13
5	Задание на строки	14
5.1	Табличная функция	14
5.1.1	Задание	14
5.1.2	Теоретические сведения	14
5.1.3	Проектирование	14
5.1.4	Описание тестового стенда и методики тестирования	15
5.1.5	Тестовый план и результаты тестирования	15
5.1.6	Выводы	15

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 (printf,scanf) для ввода и вывода информации, образцы которых находятся в `<stdio.h>`
- функция (getch) 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 (функции (printf,scanf), функция (getch)). Знать что такое цикл и для чего он нужен, в данном задании я использовал цикл "for" для создания циклов.

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.

$$const int n = 3; const int c[n] = 5, 2, 1; \quad (1.1)$$

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 отсутствовало. Для статического анализа был использован cppcheckgui версии 1.7.1 . Ошибок и предупреждений не было.

1.1.5 Тестовый план и результаты тестирования

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

1. входные данные задавал случайно - 8423
ожидаемы значения - "Программа сосчитает наименьшее кол-во монет для достижения заданных данных"
полученные - 5 rub. - 1684 coin 2 rub. - 1 coin 1 rub. - 1 coin

1.1.6 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 (printf,scanf)
- конструкция "if"использующая оператор "if"реализующий выполнение определённых команд
- конструкция "while"использующая цикл "while"благодаря которому мы задаём условие до выполнения тела цикла.

Для решения данной задачи необходимо было уметь считывать и записывать информацию в файл в этом мне помогли знания в области библиотека (ввод-вывод) включающие в себя такие функции как (fopen,fclose) также нужно было знать функцию (fmode) с помощью которой мы указывали что нужно сделать с файлом , а также уметь корректировать текст с помощью функции (gets), которая считывает строку стандартного ввода и помещает ее в буфер и функция (puts) которая выводит передаваемый ей аргумент на экран, завершая вывод переходом на новую строку.

Требовалось считывать текст из файла, делать проверку на наличие определенных символов и записывать исправленный текст в файл, мне нужно было как-то избавляться от не нужного мусора, функция (remove) удаляла файлы, имена которых указаны в параметре fname.

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

Ручное тестирование отсутствовало.

2.1.5 Тестовый план и результаты тестирования

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

1. входные данные - \$ Я не еврей
ожидаемы значения - программа создаст красную строку
полученные - "Я еврей"
2. входные данные - # Собака
ожидаемые значения - программа удалит собаку, бедная собака(
полученные -
3. входные данные - @ Кошка и остатки от собаки
ожидаемые значения - программа удалила собаку, теперь ей нужно удалить кошку, чтобы прервать пищевую цепочку и завоевать мир
полученные -

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                 else
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] != '?' && s[i + 1] != '\0')
86             i++;
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:

- функции (printf,scanf) для ввода и вывода информации, образцы которых находятся в <stdio.h>
- оператор "if"обеспечивающий выборочное выполнение отдельных участков кода
- целочисленный тип данных "int"для представления целых чисел

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 отсутствовало. Для статического анализа был использован cppcheckgui. Ошибок и предупреждений не было.

3.1.5 Тестовый план и результаты тестирования

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

1. входные данные - [4 6 1 3 4 2]
ожидаемы значения - программа определит от кого исходит угроза или между кем она возникла
полученные - "The threat between 1 and 2 a pair of Queens"
2. входные данные - [2 9 3 6 6 3]
ожидаемы значения - программа определит от кого исходит угроза или между кем она возникла
полученные - "The threat between 2 and 3 a pair of Queens"
3. входные данные - [4 4 5 6 7 4]
ожидаемы значения - программа определит от кого исходит угроза или между кем она возникла
полученные - "The threat between 1 and 3 a pair of Queens"
4. входные данные - [2 9 3 7 4 4]
ожидаемы значения - программа определит что угроза отсутствует
полученные - "There's no threat"

3.1.6 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
12    if ((x1==x2)|| (y1==y2)|| (abs(x1-x2)==abs(y1-y2)))
13    {
14        printf("The threat between 1 and 2 a pair of Queens\n"
15              );
16        k++;
17    }
18    if ((x1==x3)|| (y1==y3)|| (abs(x1-x3)==abs(y1-y3)))
19    {
20        printf("The threat between 1 and 3 a pair of Queens\n"
21              );
22        k++;
23    }
24    if ((x2==x3)|| (y2==y3)|| (abs(x2-x3)==abs(y2-y3)))
25    {
26        printf("The threat between 2 and 3 a pair of Queens\n"
27              );
28        k++;
29    }
30    if (k==0) printf("There's no threat\n");
31    return 0;
32 }
33
34 // [4 6 1 3 4 2] - "The threat between 1 and 2 a pair of
35 //               Queens"
36 // [2 9 3 6 6 3] - "The threat between 2 and 3 a pair of
37 //               Queens"
38 // [4 4 5 6 7 4] - "The threat between 1 and 3 a pair of
39 //               Queens"
40 // [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 отсутствовало. Для статического анализа был использован crrcheckgui. Ошибок и предупреждений не было.

4.1.5 Тестовый план и результаты тестирования

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

1. входные данные - 1 2 3
ожидаемы значения - программа не найдёт корней уравнения
полученные - "No roots"
2. входные данные - 1 4 3
ожидаемы значения - программа выведет найденные корни
полученные - $e = -1$ $f = -3$

4.1.6 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 отсутствовало.

5.1.5 Тестовый план и результаты тестирования

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

1. входные данные - подобрал нелогичное уравнение так как в задании этого не требовалось $y = (3x-8) + (23x+x(9x-13))$

ожидаемые значения - программа должна была определить и разбить скобки по уровням, скорректировать их и произвести замену нужных

полученные - $y = [3x-8] + [23x+x(9x-13)]$

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 }

```

Глава 6

Задание на классы

6.1 Очередь

6.1.1 Задание 1

Реализовать класс ОЧЕРЕДЬ (целых чисел, неограниченного размера). Требуемые методы: конструктор, деструктор, копирование, встать в очередь, выйти из очереди.

6.1.2 Теоретические сведения

Было использовано:

- потоки ввода и вывода информации (iostream)
- конструкция "if"
- циклы "while" и "for"
- "class" объявляет тип класса
- "void" тип указателя
- оператор "break" заканчивающий выполнение ближайшего внешнего цикла
- оператор "continue" обеспечивающий передачу управления управляющему выражению
- оператор "delete" отменяющий выделение блока памяти.

Для решения данной задачи необходимо было умение работать с классами и потоками ввода и вывода, а также уметь пользоваться операторами (break,continue,delete) и уметь объявлять тип класса. Знания циклов также необходимо для решения поставленной задачи.

6.1.3 Проектирование

Была использована только одна функция main.cpp так как программа была выполнена в Visual Studio 2013.

6.1.4 Описание тестового стенда и методики тестирования

Использовался QtCreator с GCC компилятором
Операционная система: Windows 10

Ручное тестирование отсутствовало.

6.1.5 Тестовый план и результаты тестирования

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

входные данные - [1 1 1 2 3 4 0]

ожидаемы значения - при нажатии 1- программа добавляет элемент, при нажатии 2 она просматривает очередь, при нажатии 3- удаляется элемент, при нажатии 4 производится поиск элемента, при нажатие 0 программа завершается. Задаём a,b,c (добавляем переменные, после чего удаляем одну из них на выбор, я выбрал (b-2). После удаление ищем оставшиеся элементы и завершаем процесс.

полученные - из всех элементов , остались только a,c.

6.1.6 Выводы

В ходе написания программы возникло очень много трудностей, так как class - это сложная в понимании для меня тема.

Листинги

```
1 #include <iostream>
2 #include <windows.h>
3
4
```

```

5 using namespace std;
6
7 class Node
8 {
9 public:
10     int number;
11     Node* last;
12     Node* next;
13 };
14
15 void english(const char*eng);
16
17 int main()
18 {
19     Node* head = NULL;
20     Node* tail = NULL;
21     Node* ptrLast = NULL;
22     short action = -1;
23
24     while(1)
25     {
26         english("1. Add item\n");
27         english("2. View queue\n");
28         english("3. Remove an item\n");
29         english("4. Search item\n");
30         english("0. Exit\n");
31         english("Your choice:");
32         cin>>action;
33
34         if (action == 0)
35         {
36             system("CLS");
37             break;
38         }
39
40         if (action == 1)
41         {
42             system("CLS");
43             int numb = -1;
44             english("Enter the number: ");
45             cin>>numb;
46             Node* ptr = new Node;
47             ptr->number = numb;
48             ptr->next = NULL;
49             tail = ptr;
50             if (head == NULL)
51             {
52                 head = ptr;
53                 ptrLast = ptr;

```

```

54         ptr->last = NULL;
55         system("CLS");
56         continue;
57     }
58     ptr->last = ptrLast;
59     ptrLast->next = ptr;
60     ptrLast = ptr;
61     system("CLS");
62     continue;
63 }
64
65 if (action == 2)
66 {
67     system("CLS");
68     Node* ptr = NULL;
69     if (head == NULL)
70     {
71         english("\t Queue is empty \n");
72         system("PAUSE");
73         system("CLS");
74         continue;
75     }
76     english("Queue\n");
77     ptr = tail;
78     while (1)
79     {
80         cout<<ptr->number<<" ";
81         if (ptr->last == 0)
82             break;
83         ptr = ptr->last;
84     }
85     cout<<"\n";
86     system("PAUSE");
87     system("CLS");
88     continue;
89 }
90
91 if (action == 3)
92 {
93     system("CLS");
94     Node* ptrDelete = NULL;
95     if (head == NULL)
96     {
97         english("\t Queue is empty \n");
98         system("PAUSE");
99         system("CLS");
100        continue;
101    }
102    if (head->next == NULL)

```



```

103         {
104             head = NULL;
105             tail = NULL;
106             delete tail;
107             continue;
108         }
109         ptrDelete = head;
110         head = ptrDelete->next;
111         head->last = NULL;
112         delete ptrDelete;
113         continue;
114     }
115
116     if (action == 4)
117     {
118         system("CLS");
119         Node* ptr = NULL;
120         int key = -1;
121         if (head == NULL)
122         {
123             english("\t List is empty \n");
124             system("PAUSE");
125             system("CLS");
126             continue;
127         }
128         english("Enter the element to search:");
129         cin>>key;
130         ptr = head;
131         while (1)
132         {
133             if (key == ptr->number)
134             {
135                 english("\n\t The item is found \n");
136                 break;
137             }
138             if (ptr->next == NULL)
139             {
140                 english("\n\t Item not found \n");
141                 break;
142             }
143             ptr = ptr->next;
144         }
145         system("PAUSE");
146         system("CLS");
147         continue;
148     }
149
150     if (action > 4)
151     {

```

```
152         system("CLS");
153         english("\t Error. Re-enter \n");
154         system("PAUSE");
155         system("CLS");
156         continue;
157     }
158 }
159 }
160 char word[100];
161 cout<<word;
```