

СОДЕРЖАНИЕ

Практическая работа №1 «Основы программирования на с/с++»	2
Практическая работа №2 «Циклы и ветвления».....	11
Практическая работа №3 «Структуры и перечисления»	19
Практическая работа №4 «Функции»	26

ПРАКТИЧЕСКАЯ РАБОТА №1 «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА C/C++»

Цель работы: получить практические навыки в использовании в языках C и C++ для создания программ.

Задание №1: используя средства языка C и C++ (printf(), cout) напишите программу, выводящую на экран ваше любимое стихотворение. Для разбиения на строки используйте подходящую управляющую последовательность.

Решение:

```
L1 > task1.cpp > main()
1  #include <iostream>
2  #include <vector>
3  #include <string>
4
5  using namespace std;
6
7  int main()
8  {
9      vector<string> msg
10     {
11         "Унылая пора! Очей очарованье!",
12         "Люблю я пышное природы увяданье,",
13         "В багрец и в золото одетые леса,",
14         "В их сенях ветра шум и свежее дыханье,",
15         "И мглой волнистою покрыты небеса,",
16         "И редкий солнца луч, и первые морозы,",
17         "И отдаленные седой зимы угрозы.",
18     };
19
20     for (const auto& word : msg)
21     {
22         cout << word << "\n";
23     }
24 }
```

Рисунок 1 – L1 Task1

Результат работы программы:

```
PS C:\Users\vipbo\source\c++> & 'c:\Use
rosoft-MIEngine-In-xxkyd0do.0sw' '--stdo
zplm2ee.dpv' '--dbgExe=C:\msys64\ucrt64\
Унылая пора! Очей очарованье!
Люблю я пышное природы увяданье,
В багрец и в золото одетые леса,
В их сенях ветра шум и свежее дыхание,
И мглой волнистою покрыты небеса,
И редкий солнца луч, и первые морозы,
И отдаленные седой зимы угрозы.
PS C:\Users\vipbo\source\c++> █
```

Рисунок 2 – результат L1 Task1

Блок-схема:

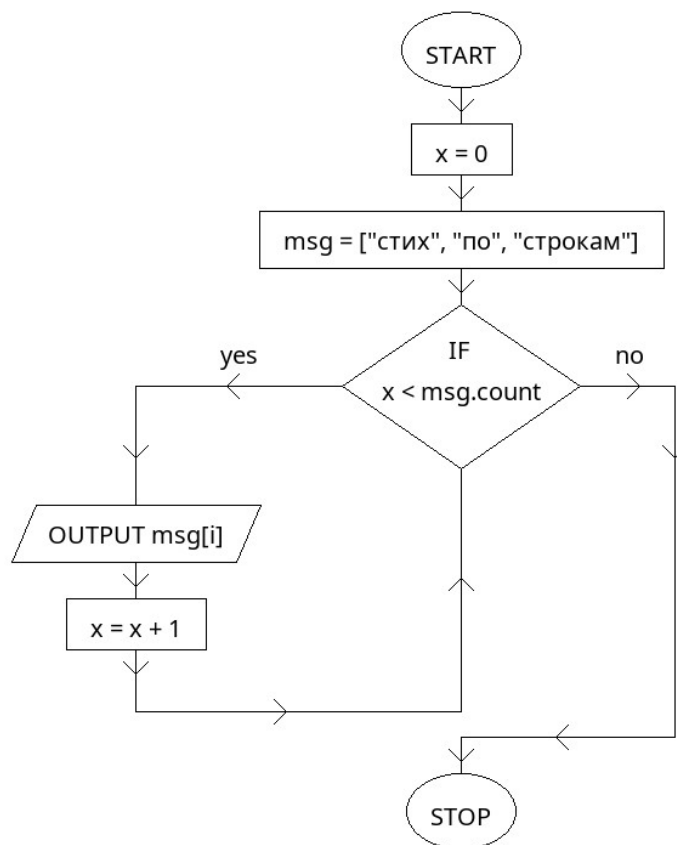


Рисунок 3 – схема L1 Task1

Задание №2: напишите программу, запрашивающую температуру в градусах по Цельсию и отображающую ее эквивалент по Фаренгейту.

Решение:

```
L1 > task2.cpp > main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main()
7  {
8      double tc, tf;
9
10     cout << "Введите температуру (Цельсий): ";
11     cin >> tc;
12
13     tf = (tc * 9/5) + 32;
14
15     cout << "Температура в градусах Фаренгейта: " << tf;
16 }
```

Рисунок 4 – L1 Task2

Результат работы программы:

```
PS C:\Users\vipbo\source\c++> & 'c:\Users\vipbo\source\c++\rossoft-MIEngine-In-4fa2un2u.nsm' '--stdout efmjdhx.eei' '--dbgExe=C:\msys64\ucrt64\b
Введите температуру (Цельсий): 100
Температура в градусах Фаренгейта: 212
PS C:\Users\vipbo\source\c++> █
```

Рисунок 5 – результат L1 Task2

Блок-схема:

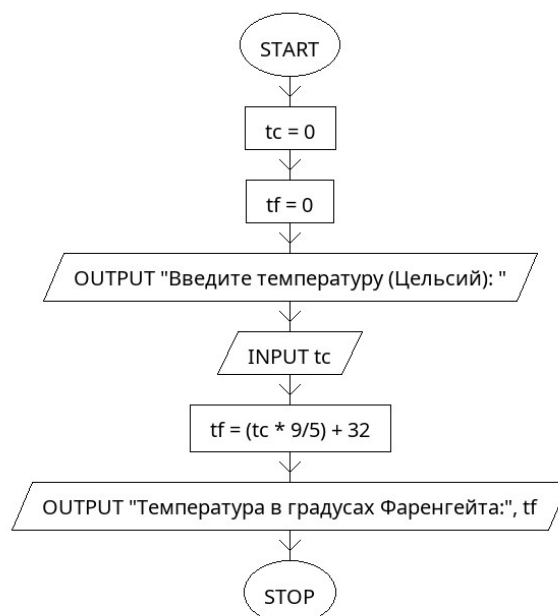


Рисунок 6 – схема L2

Задание №3: напишите программу, запрашивающую у пользователя значения двух дробей, а затем выводящую результат, также записанный в форме дроби.

Решение:

```
L1 > task3.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int a, b, c, d;
8      char dummychar;
9
10     cout << "Введите первую дробь в формате a/b: ";
11     cin >> a >> dummychar >> b;
12
13     cout << "Введите вторую дробь в формате c/d: ";
14     cin >> c >> dummychar >> d;
15
16     int numerator = a * d + b * c;
17     int denominator = b * d;
18
19     std::cout << "Сумма равна " << numerator << "/" << denominator;
20 }
21
```

Рисунок 7 - L1 Task3

Результат работы программы:

```
PS C:\Users\vipbo\source\c++> & 'c:\Users\vipbo\source\c++\rossoft-MIEngine-In-xvelrree.dzh' '--stdout x430gyl.4yc' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe'
Введите первую дробь в формате a/b: 1/2
Введите вторую дробь в формате c/d: 2/5
Сумма равна 9/10
PS C:\Users\vipbo\source\c++>
```

Рисунок 8 – результат L1 Task3

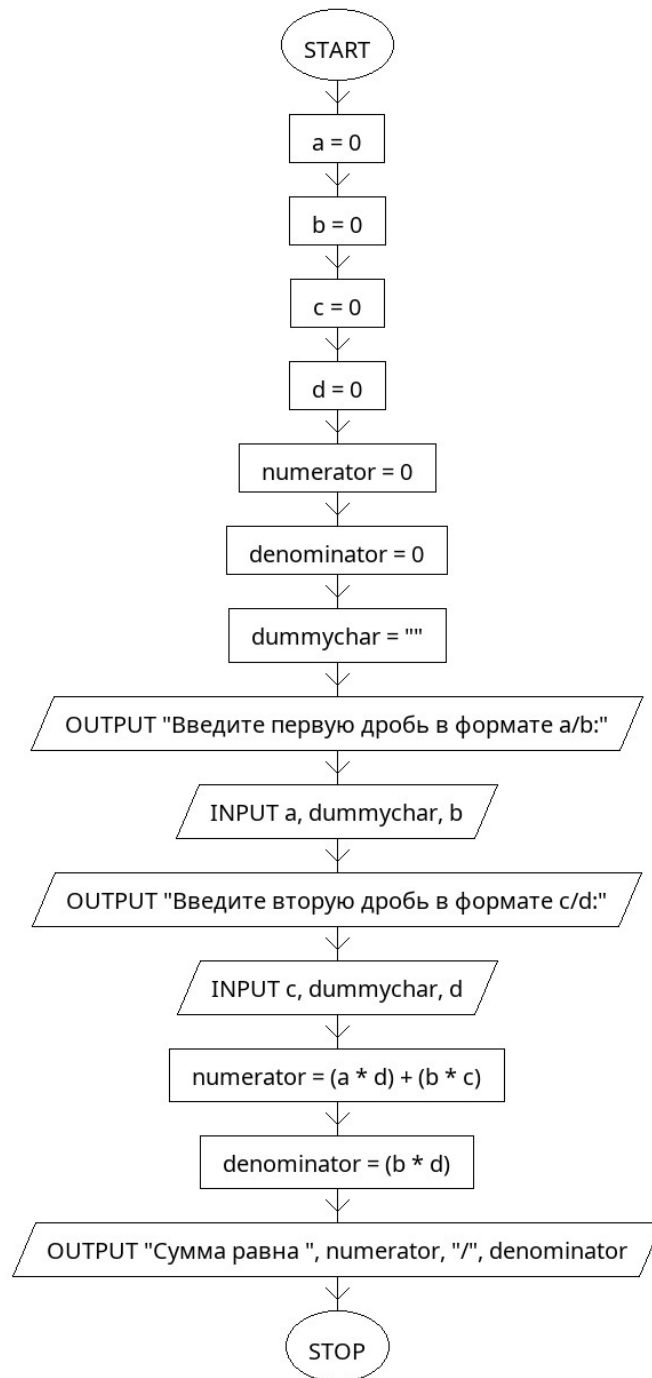


Рисунок 9 – L1 Task3

Задание №4: напишите программу, которая будет преобразовывать сумму, записанную в старом формате (фунты, шиллинги, пенсы), в новый формат (фунты, пенсы).

Решение:

```

L1 > task4.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      double pound, shilling, pence;
8
9      cout << "Введите количество фунтов: ";
10     cin >> pound;
11     cout << "Введите количество шиллингов: ";
12     cin >> shilling;
13     cout << "Введите количество пенсов: ";
14     cin >> pence;
15
16     double decimal_pounds = (pound * 240 + shilling * 12 + pence) / 240;
17
18     cout << "Десятичные фунты: £" << decimal_pounds;
19 }
20

```

Рисунок 10 – L1 Task4

Результат работы программы:

```

PS C:\Users\vipbo\source\c++> & 'c:\Users\vipbo\source\c++\task4.exe' '--stdout=ghx0dt5.qnf' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe'
Введите количество фунтов: 7
Введите количество шиллингов: 17
Введите количество пенсов: 19
Десятичные фунты: £7.92917
PS C:\Users\vipbo\source\c++>

```

Рисунок 11 – результат L1 Task4

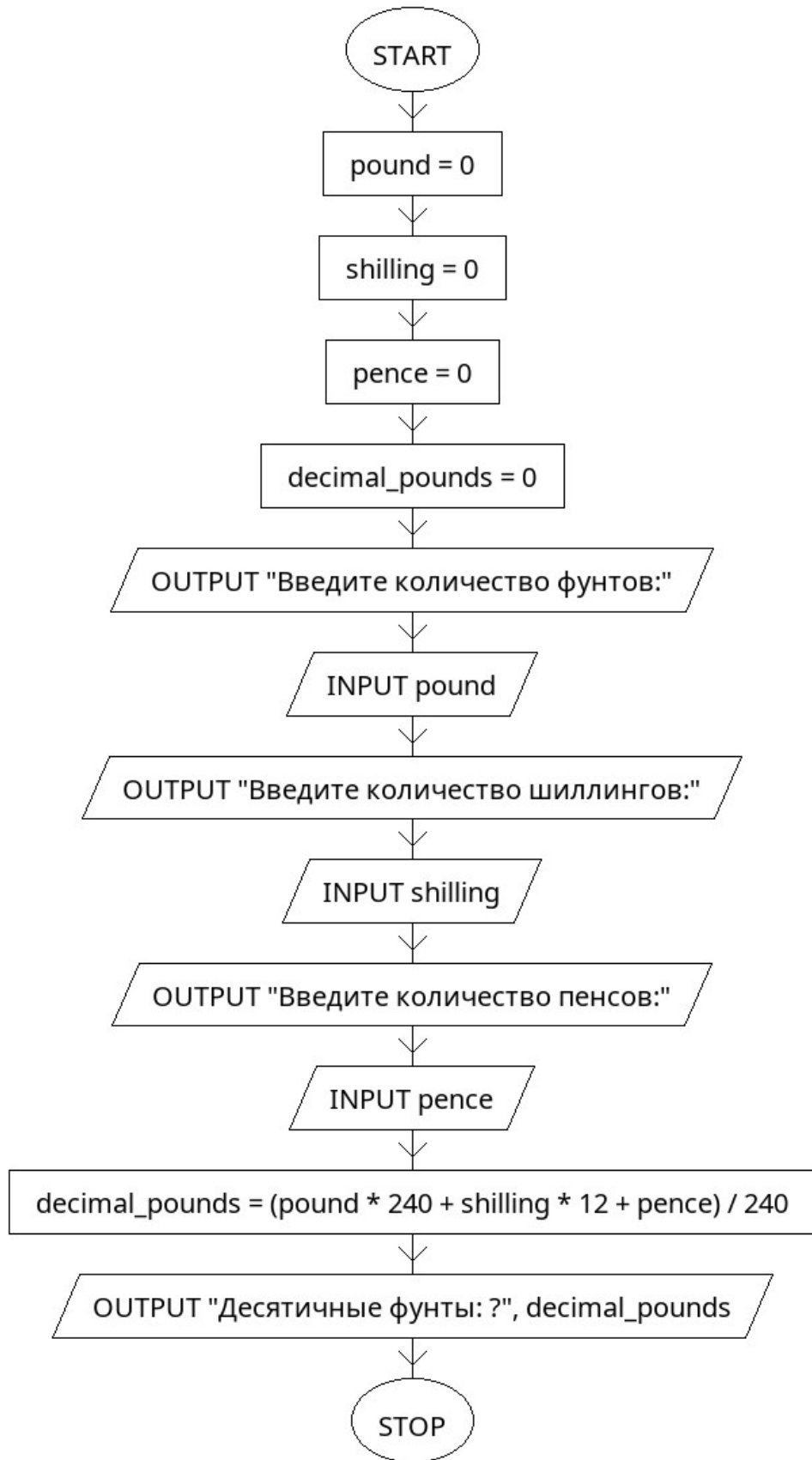


Рисунок 12 – схема L1 Task4

Задание №5: напишите программу, выполняющую действия, обратные тем, которые описаны в предыдущем упражнении 4, то есть запрашивающую у пользователя сумму, указанную в десятичных фунтах, и переводящую ее в старую систему.

```
L1 > task5.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      double decimalPounds;
8      cout << "Введите число десятичных фунтов: ";
9      cin >> decimalPounds;
10
11     int pounds = static_cast<int>(decimalPounds);
12     decimalPounds -= pounds;
13     int shillings = static_cast<int>(decimalPounds * 20);
14     decimalPounds -= shillings / 20;
15     int pence = static_cast<int>(decimalPounds * 240);
16
17     cout << "Эквивалентная сумма в старой форме записи: £"
18         << pounds << "." << shillings << "." << pence;
19 }
20
```

Рисунок 13 – L1 Task5

Результат работы программы:

```
PS C:\Users\vipbo\source\c++> & 'c:\Users\vipbo\.vsco
rosoft-MIEngine-In-x2uzxffl.4ni' '--stdout=Microsoft-M
bkzkabx.01t' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '
Введите число десятичных фунтов: 3.51
Эквивалентная сумма в старой форме записи: £3.10.122
PS C:\Users\vipbo\source\c++> █
```

Рисунок 14 – результат L1 Task5

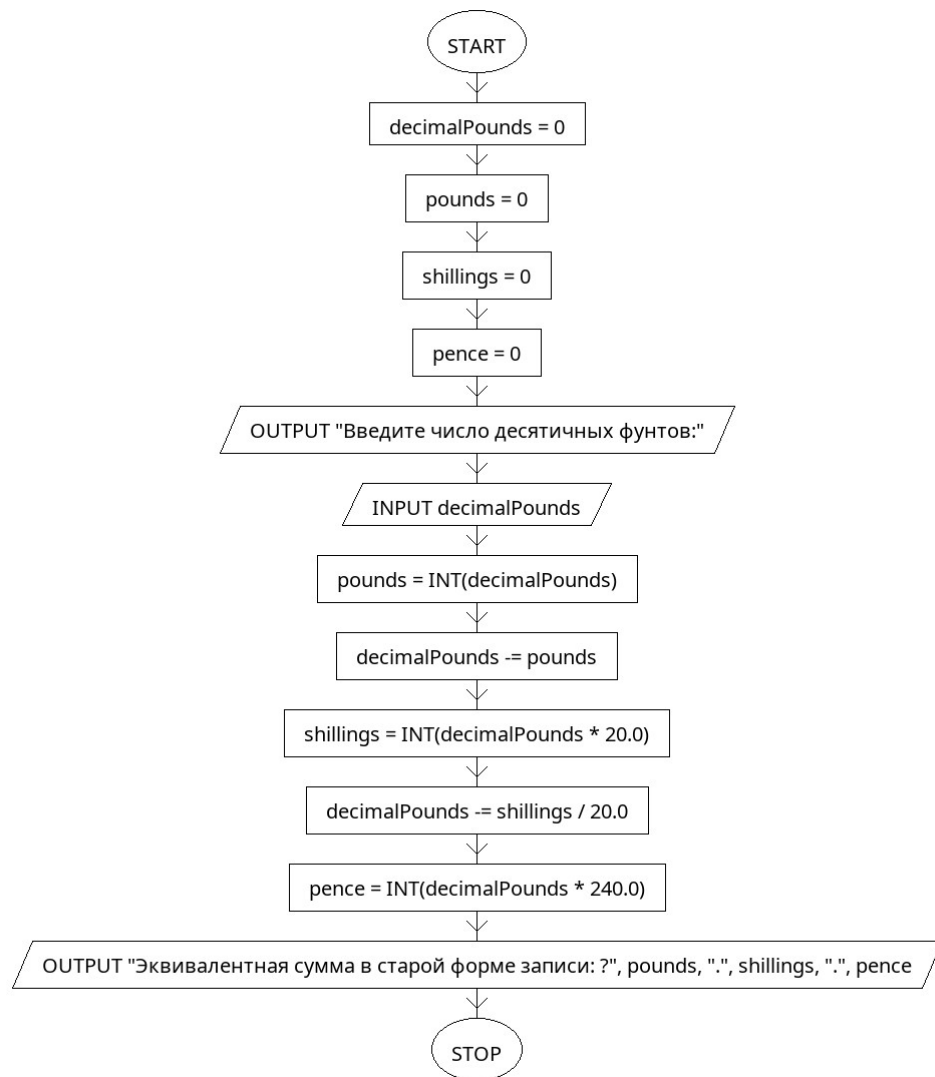


Рисунок 15 – схема L1 Task5

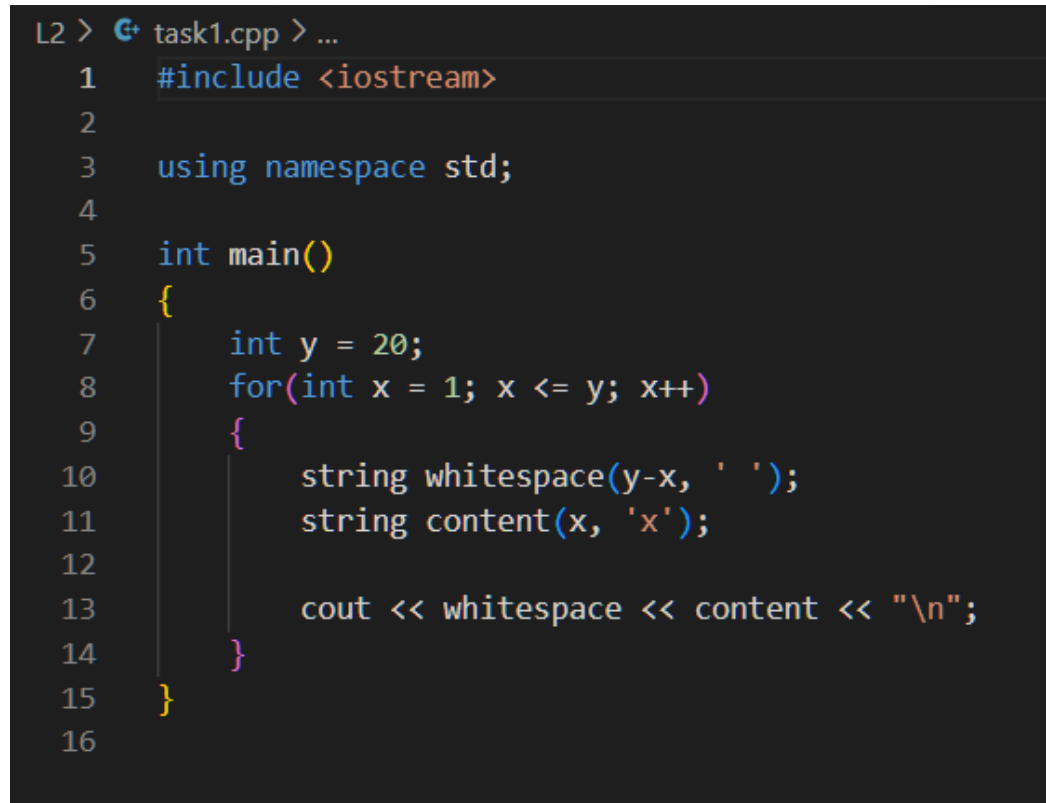
Вывод: в ходе практической работы были получены практические навыки в использовании языка с++ за счет решения задач.

ПРАКТИЧЕСКАЯ РАБОТА №2 «ЦИКЛЫ И ВЕТВЛЕНИЯ»

Цель работы: получить практические навыки в создании программ с циклами и ветвлениями.

Задание №1: при помощи цикла for изобразите на экране пирамиду из символов 'X'.

Решение:



```
L2 > task1.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int y = 20;
8      for(int x = 1; x <= y; x++)
9      {
10         string whitespace(y-x, ' ');
11         string content(x, 'x');
12
13         cout << whitespace << content << "\n";
14     }
15 }
16
```

Рисунок 16 – L2 Task1

Результат работы программы:

[illegible]

Рисунок 17 – результат L2 Task1

Задание №2: напишите программу вычисления факториала таким образом, чтобы она циклически запрашивала ввод пользователем числа и вычисляла его факториал, пока пользователь не введет 0. В этом случае программа должна завершиться. Вычисление должно строиться на основе использования цикла.

```

L2 > task2.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, result;
8
9      while (true)
10     {
11         result = 1; n = 0;
12
13         cout << "Введите 0 чтобы завершить программу\n";
14         cout << "Введите число: ";
15
16         cin >> n;
17
18         if(n == 0) break;
19
20         for(int i = n; i > 0; i--)
21             result *= i;
22
23         cout << "Факториал числа: " << result << "\n";
24     }
25 }
26

```

Рисунок 18 – L2 Task2

Результат работы программы:

```

PS C:\Users\vipbo\source\c++> & 'c:\p
rosoft-MIEngine-In-sgpnap5.f3t' '--s
e2r2bgo.d1p' '--dbgExe=C:\msys64\ucrt
Введите 0 чтобы завершить программу
Введите число: 3
Факториал числа: 6
Введите 0 чтобы завершить программу
Введите число: 6
Факториал числа: 720
Введите 0 чтобы завершить программу
Введите число: 10
Факториал числа: 3628800
Введите 0 чтобы завершить программу
Введите число: 0
PS C:\Users\vipbo\source\c++>

```

Рисунок 19 - результат L2 Task2

Задание №3: Напишите программу, рассчитывающую сумму денег, которые вы получите при вложении начальной суммы с фиксированной процентной ставкой дохода через определенное количество лет. Пользователь должен вводить с клавиатуры начальный вклад, число лет и процентную ставку.

Решение:

```
L2 > task3.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      double start_c, years, percent, sum;
8
9      cout << "Введите начальный вклад: ";
10     cin >> start_c;
11
12     cout << "Введите число лет: ";
13     cin >> years;
14
15     cout << "Введите процентную ставку: ";
16     cin >> percent;
17
18     percent /= 100;
19
20     sum = start_c;
21
22     for(int i = 0; i < years; i++){
23         sum += (sum * percent);
24         cout << "В конце " << i+1 << " года вы получите " << sum << " долларов. \n";
25     }
26
27     cout << "Через " << years << " лет вы получите " << sum << " доллара.";
28 }
29
```

Рисунок 20 – L2 Task3

Результат работы программы:

```
1303043.0yu --ubgxe-c. (msys64\usr\bin\ga
Введите начальный вклад: 3000
Введите число лет: 10
Введите процентную ставку: 5.5
В конце 1-го года вы получите 3165долларов.
В конце 2-го года вы получите 3339.07долларов.
В конце 3-го года вы получите 3522.72долларов.
В конце 4-го года вы получите 3716.47долларов.
В конце 5-го года вы получите 3920.88долларов.
В конце 6-го года вы получите 4136.53долларов.
В конце 7-го года вы получите 4364.04долларов.
В конце 8-го года вы получите 4604.06долларов.
В конце 9-го года вы получите 4857.28долларов.
В конце 10-го года вы получите 5124.43долларов.
Через 10 лет вы получите 5124.43 доллара.
PS C:\Users\vipbo\source\c++>
```

Рисунок 21 – результат L2 Task3

Задание №4: напишите программу, рассчитывающую сумму денег, которые придется выплатить при покупке квартиры в ипотеку с фиксированной процентной ставкой к моменту окончания всех платежей. Пользователь должен вводить с клавиатуры стоимость квартиры, начальный взнос, число лет ипотеки и процентную ставку. На экран вывести объемы платежей по годам и общую сумму выплат.

Решение:

```
L2 > task4.cpp > main()
3
4 using namespace std;
5
6 int main() {
7     double apartmentCost, initialPayment, annualInterestRate;
8     int years;
9
10    cout << "Введите стоимость квартиры: ";
11    cin >> apartmentCost;
12
13    cout << "Введите начальный взнос: ";
14    cin >> initialPayment;
15
16    cout << "Введите число лет ипотеки: ";
17    cin >> years;
18
19    cout << "Введите годовую процентную ставку: ";
20    cin >> annualInterestRate;
21
22    double loanAmount = apartmentCost - initialPayment;
23    double monthlyInterestRate = annualInterestRate / 12 / 100;
24    double totalMonths = years * 12;
25    double totalRate = pow((1 + monthlyInterestRate), totalMonths);
26    double monthlyPayment = loanAmount * monthlyInterestRate * totalRate / (totalRate - 1);
27
28    cout << "Ежемесячный платеж: " << monthlyPayment << endl;
29
30    double remainingDebt = loanAmount;
31
32    for(int month = 1; month <= totalMonths; ++month)
33    {
34        double percentagePart = remainingDebt * monthlyInterestRate;
35        double mainPart = monthlyPayment - percentagePart;
36        remainingDebt -= mainPart;
37
38        if (month % 12 == 1) {
39            int year = (month - 1) / 12 + 1;
40            cout << "Год " << year << ": " << percentagePart << " (проценты) | "
41                << mainPart << " (основной долг) | Остаток: " << remainingDebt << " рублей\n";
42        }
43    }
44
45    cout << "Общая сумма выплат: " << monthlyPayment * totalMonths;
46
47 }
```

Рисунок 22 L2 Task4

Результат работы программы:

```
Введите стоимость квартиры: 2000000
Введите начальный взнос: 500000
Введите число лет ипотеки: 20
Введите годовую процентную ставку: 5.5
Ежемесячный платеж: 10318.3
Год 1: 6875 (проценты) | 3443.31 (основной долг) | Остаток: 1.49656e+06 рублей
Год 2: 6680.77 (проценты) | 3637.54 (основной долг) | Остаток: 1.45399e+06 рублей
Год 3: 6475.58 (проценты) | 3842.73 (основной долг) | Остаток: 1.40901e+06 рублей
Год 4: 6258.82 (проценты) | 4059.49 (основной долг) | Остаток: 1.3615e+06 рублей
Год 5: 6029.84 (проценты) | 4288.47 (основной долг) | Остаток: 1.31131e+06 рублей
Год 6: 5787.93 (проценты) | 4530.38 (основной долг) | Остаток: 1.25829e+06 рублей
Год 7: 5532.39 (проценты) | 4785.92 (основной долг) | Остаток: 1.20228e+06 рублей
Год 8: 5262.42 (проценты) | 5055.89 (основной долг) | Остаток: 1.14311e+06 рублей
Год 9: 4977.23 (проценты) | 5341.08 (основной долг) | Остаток: 1.0806e+06 рублей
Год 10: 4675.95 (проценты) | 5642.36 (основной долг) | Остаток: 1.01457e+06 рублей
Год 11: 4357.68 (проценты) | 5960.63 (основной долг) | Остаток: 944805 рублей
Год 12: 4021.45 (проценты) | 6296.86 (основной долг) | Остаток: 871111 рублей
Год 13: 3666.26 (проценты) | 6652.05 (основной долг) | Остаток: 793259 рублей
Год 14: 3291.03 (проценты) | 7027.28 (основной долг) | Остаток: 711016 рублей
Год 15: 2894.64 (проценты) | 7423.67 (основной долг) | Остаток: 624133 рублей
Год 16: 2475.88 (проценты) | 7842.43 (основной долг) | Остаток: 532350 рублей
Год 17: 2033.51 (проценты) | 8284.8 (основной долг) | Остаток: 435390 рублей
Год 18: 1566.18 (проценты) | 8752.13 (основной долг) | Остаток: 332960 рублей
Год 19: 1072.49 (проценты) | 9245.82 (основной долг) | Остаток: 224753 рублей
Год 20: 550.956 (проценты) | 9767.35 (основной долг) | Остаток: 110441 рублей
Общая сумма выплат: 2.47639e+06
```

Рисунок 23 - результат L2 Task4

Задание №5: напишите программу, которая циклически будет запрашивать ввод пользователем двух денежных сумм, выраженных в фунтах, шиллингах и пенсах (см. упражнения предыдущей ЛР). Программа должна складывать введенные суммы и выводить на экран результат, также выраженный в фунтах, шиллингах и пенсах. После каждой итерации программа должна спрашивать пользователя, желает ли он продолжать работу программы

Решение:


```

L2 > task5.cpp > calculate()
5
6 bool next()
7 {
8     string s;
9     while (true)
10    {
11        cout << "Продолжить (y/n)?";
12        cin >> s;
13        if (s == "y") return true;
14        if (s == "n") return false;
15        cout << "Неверный формат\n";
16    }
17 }
18
19 void calculate()
20 {
21     int poundsA, shillingsA, penceA;
22     int poundsB, shillingsB, penceB;
23
24     std::cout << "Введите первое значение: (фунты шиллинги пенсы): ";
25     std::cin >> poundsA >> shillingsA >> penceA;
26
27     std::cout << "Введите второе значение: (фунты шиллинги пенсы): ";
28     std::cin >> poundsB >> shillingsB >> penceB;
29
30     int totalPounds = poundsA + poundsB;
31     int totalShillings = shillingsA + shillingsB;
32     int totalPence = penceA + penceB;
33
34     if (totalPence >= 12) {
35         totalShillings += totalPence / 12;
36         totalPence %= 12;
37     }
38
39     if (totalShillings >= 20) {
40         totalPounds += totalShillings / 20;
41         totalShillings %= 20;
42     }
43
44     cout << "Итого: " << totalPounds << " фунт "
45         << totalShillings << " шиллингов "
46         << totalPence << " пенс" << endl;
47 }

```

Рисунок 24 – L2 Task5 Методы next и calculate

```

L2 > task5.cpp > calculate()
48
49 int main()
50 {
51     do
52     {
53         calculate();
54     } while (next());
55 }

```

Рисунок 25 - L2 Task5

Результат работы программы:

```
Введите первое значение: (фунты шиллинги пенсы): 5 10 6
Введите второе значение: (фунты шиллинги пенсы): 3 2 6
Всего: 8 фунт 13 шиллингов 0 пенс
Продолжить (y/n)?y
Введите первое значение: (фунты шиллинги пенсы): 10 13 20
Введите второе значение: (фунты шиллинги пенсы): 20 13 10
Всего: 31 фунт 8 шиллингов 6 пенс
Продолжить (y/n)?n
PS C:\Users\vipbo\source\c++>
```

Рисунок 26 – результат L2 Task5

Вывод: в ходе практической работы были получены практические навыки в создании программ с циклами и ветвлениями.

ПРАКТИЧЕСКАЯ РАБОТА №3 «СТРУКТУРЫ И ПЕРЕЧИСЛЕНИЯ»

Цель работы: получить практические навыки в создании программ, использующих структуры и перечисления.

Задание №1: создайте структуру с именем `employee`, содержащую два поля: номер сотрудника типа `int` и величину его зарплаты в рублях типа `float`. Запросите с клавиатуры данные о трех сотрудниках, сохраните их в трех структурных переменных типа `employee` и выведите информацию о каждом из сотрудников на экран.

Решение:

```
L3 > task1.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <Windows.h>
4
5  using namespace std;
6
7  struct Employee
8  {
9      public:
10         string name;
11         float salary;
12
13         Employee()
14         {
15             cout << "Введите имя сотрудника: ";
16             cin.ignore();
17             cin >> this->name;
18
19             cout << "Введите оклад сотрудника: ";
20             cin >> this->salary;
21         }
22     };
```

Рисунок 27 – L3 Task1 (1)

```

L3 > task1.cpp > ...
23
24 int main()
25 {
26     SetConsoleCP(1251);
27     SetConsoleOutputCP(1251);
28
29     vector<Employee> office;
30
31     for (int i = 0; i < 3; i++)
32         office.push_back(Employee());
33
34     cout << endl;
35
36     for (int i = 0; i < office.size(); i++)
37         cout << "Сотрудник №" << i + 1
38             << " | Имя: " << office[i].name
39             << " | Оклад: " << office[i].salary << endl;
40 }

```

Рисунок 28 - L3 Task1 (2)

Результат выполнения программы:

```

PS C:\Users\vipbo\source\c++> & 'c:\Users\vipbo\source\c++\Microsoft-MIEngine-In-3rhxnw42.01r' '--stdout=Microsoft-1ztd3t3.jsn' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe
Введите имя сотрудника: Дана
Введите оклад сотрудника: 340000
Введите имя сотрудника: Петя
Введите оклад сотрудника: 280000
Введите имя сотрудника: Вася
Введите оклад сотрудника: 180000

Сотрудник №1 | Имя: аня | Оклад: 340000
Сотрудник №2 | Имя: Петя | Оклад: 280000
Сотрудник №3 | Имя: Вася | Оклад: 180000
PS C:\Users\vipbo\source\c++>

```

Рисунок 29 – результат L3 Task1

Задание №2: Пусть, например, в программе определен перечисляемый тип данных `etype`, отражающий должность сотрудника: `enum etype { laborer, secretary, manager, accountant, executive, researcher };`

Напишите программу, которая сначала по первой букве должности, введенной пользователем, определяет соответствующее значение переменной, помещает это значение в переменную типа `etype`, а затем

ВЫВОДИТ полностью название должности, первую букву которой ввел пользователь.

Решение: в данном решении я не использовал switch (надеюсь меня простят)

```
L3 > task2.cpp > [e] etypes
7
8  enum Etype
9  {
10     laborer,
11     secretary,
12     manager,
13     accountant,
14     executive,
15     researcher
16 };
17
18 map<char, Etype> etypes = {
19     {'l', laborer},
20     {'s', secretary},
21     {'m', manager},
22     {'a', accountant},
23     {'e', executive},
24     {'r', researcher},
25 };
26
27 string etypeNames[6] = {
28     "laborer",
29     "secretary",
30     "manager",
31     "accountant",
32     "executive",
33     "researcher",
34 };
```

Рисунок 30 – L3 Task2 данные

```
L3 > task2.cpp > main()
35
36 int main()
37 {
38     cout << "Нажмите любую клавишу для продолжения..." << endl;
39
40     char ch = _getch();
41
42     int index = etypes[ch];
43
44     if(index == 0 && ch != '1')
45     {
46         cout << "Должность начинающаяся с [" << ch << "] не найдена";
47         return 0;
48     }
49
50     cout << "Полное название должности: " << etypeNames[index] << endl;
51
52     return 0;
53 }
```

Рисунок 31 – L3 Task2 main

Результат работы программы:

```
Нажмите любую клавишу для продолжения...
Полное название должности: accountant
```

Рисунок 32 – L3 Task2

Задание №3: Вернитесь к упражнению 3 ЛР1. В этом упражнении требуется написать программу, которая хранит значения двух дробей в виде числителя и знаменателя, а затем складывает эти дроби согласно арифметическому правилу. Измените эту программу так, чтобы значения дробей хранились в структуре `fraction`, состоящей из двух полей типа `int`, предназначенных для хранения числителя и знаменателя. Все значения дробей должны храниться в переменных типа `fraction`.

Решение:

```

L3 > task3.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  struct Fraction
6  {
7      public:
8          int numerator;
9          int denominator;
10
11      Fraction()
12      {
13          char dummychar;
14          cout << "Введите дробь в формате a/b: ";
15          cin >> this->numerator >> dummychar >> this->denominator;
16      }
17  };
18
19
20 int main()
21 {
22     auto frac1 = Fraction();
23     auto frac2 = Fraction();
24
25     int numerator = frac1.numerator * frac2.denominator + frac1.denominator * frac2.numerator;
26     int denominator = frac1.denominator * frac2.denominator;
27
28     std::cout << "Сумма равна " << numerator << "/" << denominator;
29 }

```

Рисунок 33 – L3 Task3

Результат работы программы:

```

PS C:\Users\vipbo\source\c++> & 'c:\rossoft-MIEngine-In-n0bl0e3c.1au' '--
zk4fzzm.xlw' '--dbgExe=C:\msys64\uc
Введите дробь в формате a/b: 1/2
Введите дробь в формате a/b: 1/2
Сумма равна 4/4
PS C:\Users\vipbo\source\c++>

```

Рисунок 34 – результат L3 Task3

Задание №4: Создайте структуру с именем time. Три ее поля, имеющие тип int. будут называться hours, minutes и seconds. Напишите программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Программа должна хранить время в структурной переменной типа time и выводить количество секунд в введенном времени.

Решение:

```

L3 > task4.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  struct Time
6  {
7      public:
8          int hours, minutes, seconds;
9
10         Time()
11         {
12             char dummychar;
13             cout << "Введите время в формате HH:MM:SS" << endl;
14             cin >> this->hours >> dummychar >> this->minutes >> dummychar >> this->seconds;
15         }
16     };
17
18
19     int main()
20     {
21         auto time = Time();
22         int totalSeconds = time.hours * 3600 + time.minutes*60 + time.seconds;
23         cout << "Всего секунд: " << totalSeconds;
24     }

```

Рисунок 35 – L3 Task4

Результат работы программы:

```

PS C:\Users\vipbo\source\c++> & '
rossoft-MIEngine-In-sxfxl4k.ykv' '
a22ibsz.hy1' '--dbgExe=C:\msys64\u
Введите время в формате HH:MM:SS
1:1:1
Всего секунд: 3661
PS C:\Users\vipbo\source\c++>

```

Рисунок 36 – результат L3 Task4

Задание №5: Используя структуру time из предыдущего упражнения, напишите программу, которая получает от пользователя два значения времени в формате 12:59:59, сохраняет их в переменных типа struct time, затем переводит оба значения в секунды, складывает их, переводит сумму в исходный формат, сохраняет его в переменной типа time и выводит полученный результат на экран в формате 12:59:59.

Решение:


```

L3 > task5.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  struct Time
6  {
7      public:
8          int hours, minutes, seconds;
9
10         Time()
11         {
12             char dummychar;
13             cout << "Введите время в формате HH:MM:SS" << endl;
14             cin >> this->hours >> dummychar >> this->minutes >> dummychar >> this->seconds;
15         }
16
17         Time(int totalSeconds)
18         {
19             this->hours = totalSeconds / 3600;
20             this->minutes = (totalSeconds % 3600) / 60;
21             this->seconds = totalSeconds % 60;
22         }
23     };
24
25
26     int main()
27     {
28         auto time1 = Time();
29         auto time2 = Time();
30         int totalSeconds1 = time1.hours * 3600 + time1.minutes*60 + time1.seconds;
31         int totalSeconds2 = time2.hours * 3600 + time2.minutes*60 + time2.seconds;
32         int totalSeconds = totalSeconds1 + totalSeconds2;
33         auto time = Time(totalSeconds);
34
35         cout << time.hours << ":" << time.minutes << ":" << time.seconds;
36     }

```

Рисунок 37 – L3 Task5

Результат работы программы:

```

PS C:\Users\vipbo\source\c++> & 'c:\rossoft-MIEngine-In-kd2x42ex.wns' '-
dmgaltt.pcx' '--dbgExe=C:\msys64\uc
Введите время в формате HH:MM:SS
1:1:1
Введите время в формате HH:MM:SS
2:2:2
3:3:3
PS C:\Users\vipbo\source\c++>

```

Рисунок 38 – результат L3 Task5

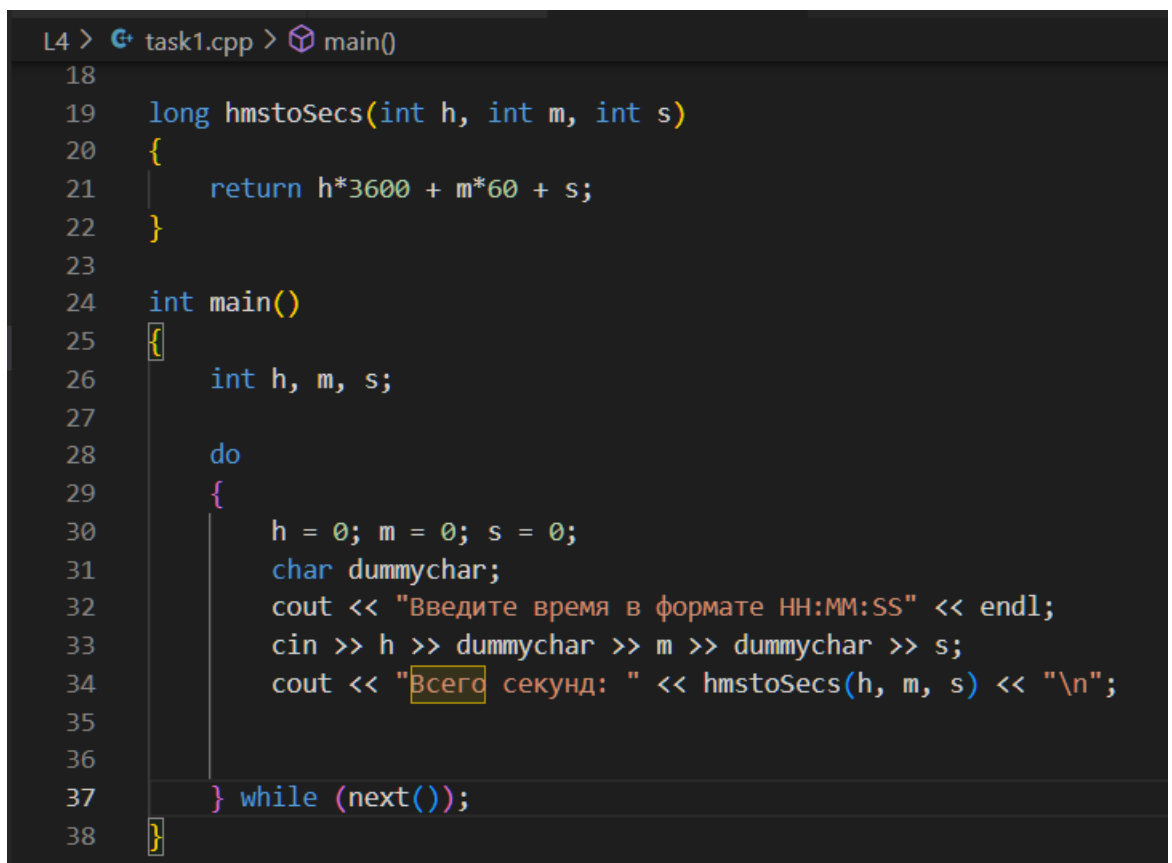
Вывод: в ходе практической работы были получены практические навыки в создании программ, использующих структуры и перечисления.

ПРАКТИЧЕСКАЯ РАБОТА №4 «ФУНКЦИИ»

Цель работы: получить практические навыки в создании функций и их последующем использовании.

Задание №1: Напишите функцию с именем `hms_to_secs()`, имеющую три аргумента типа `int`: часы, минуты и секунды. Функция должна возвращать эквивалент переданного ей временного значения в секундах (типа `long`). Создайте программу, которая будет циклически запрашивать у пользователя ввод значения часов, минут и секунд и выводить результат работы функции на экран.

Решение: функция `next()` была ранее использована в L2 Task5



```
L4 > task1.cpp > main()
18
19 long hmsToSecs(int h, int m, int s)
20 {
21     return h*3600 + m*60 + s;
22 }
23
24 int main()
25 {
26     int h, m, s;
27
28     do
29     {
30         h = 0; m = 0; s = 0;
31         char dummychar;
32         cout << "Введите время в формате HH:MM:SS" << endl;
33         cin >> h >> dummychar >> m >> dummychar >> s;
34         cout << "Всего секунд: " << hmsToSecs(h, m, s) << "\n";
35     }
36     while (next());
37 }
38
```

Рисунок 39 – L4 Task1

Результат работы программы:

```
Введите время в формате HH:MM:SS
1:1:1
Всего секунд: 3661
Продолжить (y/n)?y
Введите время в формате HH:MM:SS
2:2:2
Всего секунд: 7322
Продолжить (y/n)?d
Неверный формат
Продолжить (y/n)?n
```

Рисунок 40 – результат L4 Task1

Задание №2: модифицируйте программу, описанную в упражнении 5 предыдущей лабораторной работы «Структуры и перечисления», складывающую два структурных значения типа `time`. Теперь программа должна включать в себя две функции. Первая, `time_to_secs()`, принимает в качестве аргумента значение типа `time` и возвращает эквивалентное значение в секундах типа `long`. Вторая, `secs_to_time()`, в качестве аргумента принимает число секунд, имеющее тип `long`, а возвращает эквивалентное значение типа `time`.

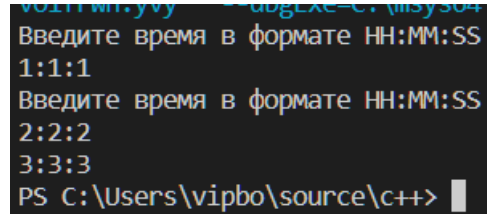
Решение:

```
L4 > task2.cpp > ...
14     cin >> this->hours >> dummychar >> this->minutes >> dummychar >> this->seconds;
15 }
16
17 Time(const int& hours, const int& minutes, const int& seconds)
18 : hours(hours), minutes(minutes), seconds(seconds) {
19 }
20
21 };
22
23 long timeToSecs(Time time)
24 {
25     return time.hours * 3600 + time.minutes*60 + time.seconds;
26 }
27
28 Time secsToTime(long seconds)
29 {
30     int hours = seconds / 3600;
31     int minutes = (seconds % 3600) / 60;
32     int seconds = seconds % 60;
33     return Time(hours, minutes, seconds);
34 }
35
36 int main()
37 {
38     auto time1 = Time();
39     auto time2 = Time();
40     int totalSeconds1 = time1.hours * 3600 + time1.minutes*60 + time1.seconds;
41     int totalSeconds2 = time2.hours * 3600 + time2.minutes*60 + time2.seconds;
42     int totalSeconds = totalSeconds1 + totalSeconds2;
43     auto time = secsToTime(totalSeconds);
44
45     cout << time.hours << ":" << time.minutes << ":" << time.seconds;
46 }
```

Рисунок 41 – L4 Task2

В данном решении используется структура из L3 Task4 с дополнительным конструктором, который принимает аргументы hours, minutes и seconds.

Результат работы программы:

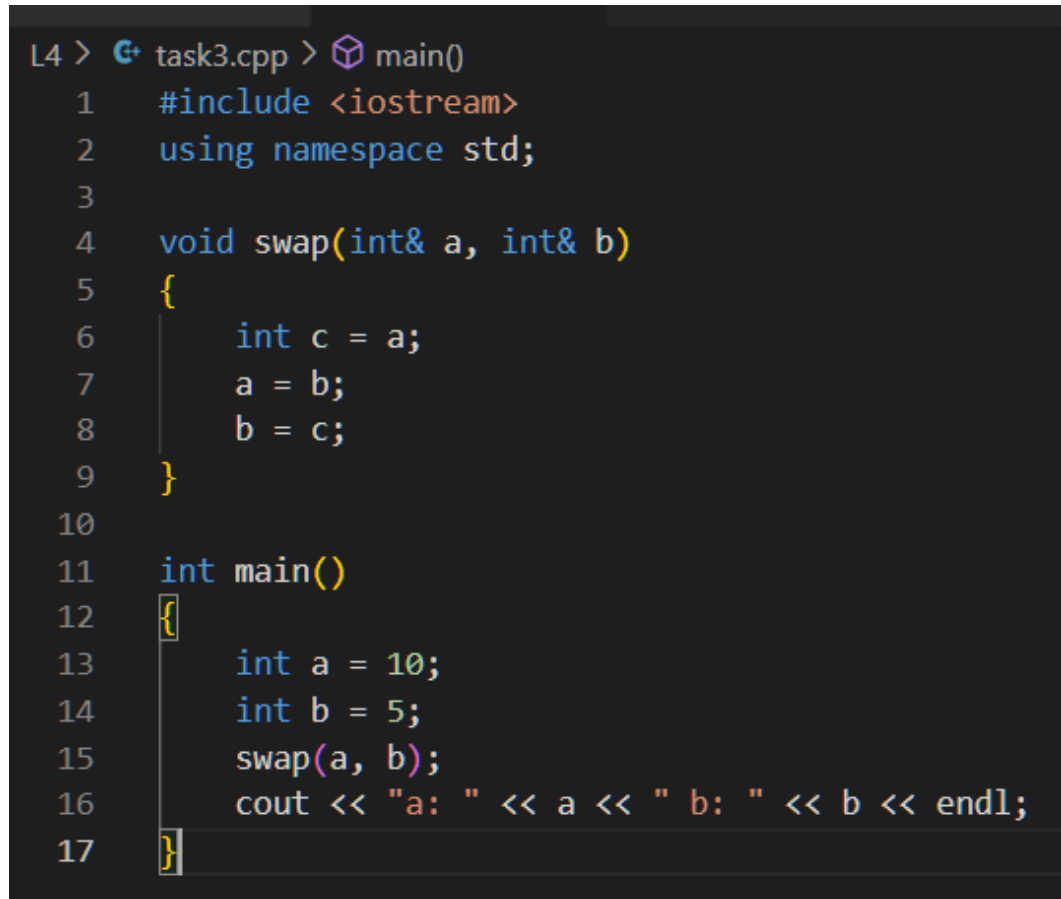


```
Введите время в формате HH:MM:SS
1:1:1
Введите время в формате HH:MM:SS
2:2:2
3:3:3
PS C:\Users\vipbo\source\c++>
```

Рисунок 42 – результат L4 Task2

Задание №3: напишите функцию с именем swap(), обменивающую значениями два своих аргумента типа int (обратите внимание, что изменяться должны значения переменных из вызывающей функции, а не локальных переменных функции). Выберите способ передачи аргументов. Напишите вызывающую программу main(), использующую данную функцию.

Решение:



```
L4 > task3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  void swap(int& a, int& b)
5  {
6      int c = a;
7      a = b;
8      b = c;
9  }
10
11 int main()
12 {
13     int a = 10;
14     int b = 5;
15     swap(a, b);
16     cout << "a: " << a << " b: " << b << endl;
17 }
```

Рисунок 43 – L3 Task3

Результат работы программы:

```
PS C:\Users\vipbo\source\c++ & 'c
rosoft-MIEngine-In-xmgmdlck.pht' '-
5t4w3m0.ge5' '--dbgExe=C:\msys64\uc
a: 5 b: 10
PS C:\Users\vipbo\source\c++
```

Рисунок 44 – результат L4 Task3

Задание №4: переработайте программу из предыдущего упражнения 3 так, чтобы функция `swar()` принимала в качестве аргументов значения типа `time` (см. упражнение 2).

Решение: данное решение было реализовано с помощью дженериков. А в структуру Time был добавлен метод toString().

```
L4 > task4.cpp > ...
20 : hours(hours), minutes(minutes), seconds(seconds) {
21 }
22
23 string toString()
24 {
25     stringstream ss;
26     ss << this->hours << ":" << this->minutes << ":" << this->seconds;
27     return ss.str();
28 }
29 };
30
31 template <typename T>
32 void mySwap(T& a, T& b) {
33     T c = a;
34     a = b;
35     b = c;
36 }
37
38 int main()
39 {
40     auto a = Time();
41     auto b = Time();
42     swap<Time>(a, b);
43     cout << "a: " << a.toString() << " b: " << b.toString() << endl;
44 }
45
```

Рисунок 45 – L4 Task4

Результат работы программы:

```
Введите время в формате HH:MM:SS
1:1:1
Введите время в формате HH:MM:SS
2:2:2
a: 2:2:2 b: 1:1:1
PS C:\Users\vipbo\source\c++>
```

Рисунок 46 – результат L4 Task4

Задание №5: напишите функцию, которая при каждом вызове будет выводить на экран количество раз, которое она вызывалась ранее. Напишите программу, которая будет вызывать данную функцию не менее 10 раз. Попробуйте реализовать данную функцию двумя различными способами: с использованием глобальной переменной и статической локальной переменной для хранения числа вызовов функции.

Решение:

```
L4 > task5.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int global = 0;
6
7  void repeateLocal()
8  {
9      static int count = 0;
10     std::cout << "Функция использующая локальную переменную вызывалась " << count << " раз" << std::endl;
11     count++;
12 }
13
14 void repeateGlobal()
15 {
16     cout << "Функция использующая глобальную переменную вызывалась " << global << " раз" << endl;
17     global++;
18 }
19
20 int main()
21 {
22     for (int i = 0; i < 10; i++) {
23         repeateLocal();
24     }
25
26     for (int i = 0; i < 10; i++) {
27         repeateGlobal();
28     }
29 }
30
```

Рисунок 47 – L4 Task5

Результат работы программы:

```
tkv1xb4.gv1 --dbgexe=C:\msys64\usr\bin\gdb.exe --interpreter=
Функция использующая локальную переменную вызывалась 0 раз
Функция использующая локальную переменную вызывалась 1 раз
Функция использующая локальную переменную вызывалась 2 раз
Функция использующая локальную переменную вызывалась 3 раз
Функция использующая локальную переменную вызывалась 4 раз
Функция использующая локальную переменную вызывалась 5 раз
Функция использующая локальную переменную вызывалась 6 раз
Функция использующая локальную переменную вызывалась 7 раз
Функция использующая локальную переменную вызывалась 8 раз
Функция использующая локальную переменную вызывалась 9 раз
Функция использующая глобальную переменную вызывалась 0 раз
Функция использующая глобальную переменную вызывалась 1 раз
Функция использующая глобальную переменную вызывалась 2 раз
Функция использующая глобальную переменную вызывалась 3 раз
Функция использующая глобальную переменную вызывалась 4 раз
Функция использующая глобальную переменную вызывалась 5 раз
Функция использующая глобальную переменную вызывалась 6 раз
Функция использующая глобальную переменную вызывалась 7 раз
Функция использующая глобальную переменную вызывалась 8 раз
Функция использующая глобальную переменную вызывалась 9 раз
PS C:\Users\vipbo\source\c++ >
```

Рисунок 48 – результат L4 Task5

Вывод: в ходе практической работы получены практические навыки в создании функций и их последующем использовании.