Бюджетное учреждение высшего образования   
Ханты-Мансийского автономного округа   
«Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**

по лабораторной работе № 1 «Линейные алгоритмы»

по дисциплине «Программирование и основы алгоритмизации»

Выполнил: Комбаев И. Д.

студент группы 609-01

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2019 г.

Цель работы:

* закрепление знаний о типах данных и операциях;
* получение практических навыков построения выражений, содержащих арифметические, логические и поразрядные операции;
* получение практических навыков использования стандартных математическиъ функций и функций стандартного ввода и вывода

Задание: Найти общее решение вычислительной задачи в соответствии с индивидуальным заданием, выданным преподавателем, разработать алгоритм решения и реализовать программу.

Вариант 29. Заданы моменты начала и конца временного интервала в часах, минутах и секундах. Найти его продолжительность в тех же единицах.

Вариант 59. Задан объем и высота резервуара, и начальная скорость вытекания жидкости. Найти время, за которое вытечет вся жидкость.

**Вариант 29**

**Формальное описание задачи.**

Стоит учесть, что 1 час = 60 минут = 3600 секунд и что 1 минута = 60 секунд. Следовательно, можно узнать общее количество времени в секундах по формуле:

Sum = H \* 3600 + M \* 60 + S

Где H – Количество часов, M – количество минут, S – количество секунд, а Sum – общее количество времени в секундах

Так как 1 целый час содержит 3600 секунд, мы можем найти количество целых часов, зная общее количество времени в секундах по формуле:

H = Sum / 3600 (1)

Применяя операцию /, мы находим количество целых часов, не учитывая остаток.

Чтобы найти количество минут, мы сначала должны остаток времени, которого не хватает для полного часа, а затем найти количество целых минут, делается это по формуле:

M = Sum % 3600 / 60 (2)

Чтобы найти секунды, мы лишь находим остаток от общего количества времени, делается это по формуле:

S = Sum % 60 (3)

Зная общее количество времени и применив формулы (1), (2), (3), мы можем выразить общее количество времени в часах, минутах и секундах.

Чтобы найти продолжительность, мы должны найти разницу между началом и концом временного интервала. Делается это путем перевода временных интервалов, выраженных в часах, минутах и секундах, в общее количество времени и последующем вычитании тех.

Time = Sum2 – Sum1

Где Time – продолжительность в секундах.

Найдя продолжительность в секундах, мы выражаем это значение в часах, минутах и секундах, использую формулы (1), (2), (3).

**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввести исходные данные – количество часов, минут, секунд первого временного интервала.
2. Вычислить количество секунд первого временного интервала.
3. Ввести исходные данные – количество часов, минут, секунд второго временного интервала.
4. Вычислить количество секунд второго временного интервала.
5. Найти разницу между первым и вторым временным интервалом, выраженную в секундах
6. Вычислить количество часов, минут, секунд из разницы между двумя временными интервалами
7. Вывести результаты на экран.

**Блок-схема алгоритма программы.**



**Листинг программы.**

1. /\* Заданы моменты начала и конца временного интервала в часах, минутах и секундах.
2. Найти его продолжительность в тех же единицах.\*/
4. #include <stdio.h>
5. #include <stdlib.h>
7. main() {
8. **unsigned** **int** h, m, s;
9. **unsigned int** sum1, sum2, different;
11. system("chcp 1251 >nul");
13. //Ввод первого временного интервала и перевод его в секунды
14. printf("Начало временного интервала в часах, минутах и секундах: ");
15. scanf("%d%d%d", &h, &m, &s);
16. sum1 = h \* 3600 + m \* 60 + s;
18. //Ввод второго временного интервала и перевод его в секунды
19. printf("Конец временного интервала в часах, минутах и секундах: ");
20. scanf("%d%d%d", &h, &m, &s);
21. sum2 = h \* 3600 + m \* 60 + s;
23. //Вычисление разницы между временными интервалами в секундах
24. different = sum2 - sum1;
26. //Перевод разницы, выраженной в секундах, в Часы, Минуты и Секунды
27. h = different / 3600;
28. m = different % 3600 / 60;
29. s = different % 60;
31. //Вывод результатов
32. printf("Продолжительность временного интервала в Часах, Минутах, Секундах: %d:%02d:%02d\n", h, m, s);
34. system("pause");
35. **return** 0;
36. }

**Пояснения к программе.**

В программе требуемые вычисления выполнены при помощи выражений:

sum1 = h \* 3600 + m \* 60 + s;

sum2 = h \* 3600 + m \* 60 + s;

different = sum2 - sum1;

h = different / 3600;

m = different % 3600 / 60;

s = different % 60;

Во всех выражениях используется одни и те же переменные h, m и s, это обусловлено тем, что после записи в них информации и её использование – данные в них больше нигде не используются, значит их можно заменить новой входящей информацией. Данный подход уменьшает количество переменных, что означает уменьшение задейственной памяти компьютера

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h. Формат вывода %d:%02d:%02d позволяет выводить секунды и минуты с незначащими нулями. Это сделано для удобства восприятия продолжительности. Так, например, продолжительность в 5 часов 5 минут и 5 секунд будет предоставлена в виде 5:05:05

Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы).

**Вариант 59**

**Формальное описание задачи.**

В качестве входных данных нам дают: Объем (Далее V), высоту (Далее H) и начальную скорость вытекания жидкости.

Заранее обговорено что:

1. Высота резервуара обозначает также и начальную высоту водяного столба.
2. Начальную скорость вытекания жидкости следует считать за Поток жидкости. (Далее q)

Вторая условность существует, потому что иначе задача не имеет решения.

Ответом на задачу служит **время, за которое вся вода вытечет из резервуара** и находится по формуле:

Где R – Радиус резервуара. a – Радиус отверстия, из которого вытекается жидкость. H – Высота. g – Ускорение свободного падения.

Нам известна высота (H) и ускорение свободного падения (g), нам нужно найти квадрат радиуса резервуара (R2) и квадрат радиус отверстия (a2), из которого вытекает жидкость.

Так как нам известен объем, высота и поток жидкости мы можем вывести формулу нахождения квадрата радиусу резервуара из формулы Объема резервуара и формулу нахождения квадрата радиуса отверстия из формулы Потока жидкости.

Подставлен выраженные (2) и (3) в формулу (1), получаем:

Заметим, что окончательная формула не содержит в себе высоту водяного столба.

**Алгоритм программы.**

Для решения данной задачи необходимо:

1. Ввести исходные данные – объем, высоту и поток жидкости
2. Вычислить время, за которое вытечет вода.
3. Вывести результаты на экран.

**Блок-схема алгоритма программы.**



**Листинг программы.**

1. /\*Задан объем и высота резервуара, и начальная скорость вытекания жидкости.
2. Найти время, за которое вытечет вся жидкость.\*/
4. #include <stdio.h>
5. #include <stdlib.h>
7. **int** main() {
9. //ввод данных
10. **double** v, v0, t;
11. system("chcp 1251 > nul");
13. printf("Введите объем, начальную скорость вытекания жидкости и высоту: ");
14. scanf("%lf%lf", &v, &v0);
16. //Вычисление времени, за которое вода вытечет из резервуара
17. t = (2 \* v) / v0;
19. //Вывод результата
20. printf("Время, за которое вытечет вода, в секундах: %lg\n", t);
21. system("pause");
22. **return** 0;
24. }

**Пояснения к программе.**

В программе требуемые вычисления выполнены при помощи выражения:

t = (2 \* v) / v0;

Стоит заметить, что в 13 строке мы просим ввести пользователя 3 значения, но обратываем всего 2. Это связано с тем, что по условии задачи нам должны дать еще и высоту, но она не участвует в вычислениях. В связи с этим было принято решение игнорировать это веденное значение.

Для организации дружественного интерфейса пользователя перед вводом данных выводится сообщение для пользователя. Ввод и вывод организован при помощи стандартных функций scanf() и printf(), для этого подключен заголовочный файл stdio.h.

Заголовочный файл stdlib.h необходим для использования функции system(), чтобы приостановить выполнение программы перед завершением (выполняется команда «pause» операционной системы).

**Выводы:**

В задаче 29 для всех переменных выбран беззнаковый целочисленный тип unsigned int, так как все величины являются целыми положительными числами, а тип unsigned int только их и поддерживает. Я не выбрал тип unsigned long, так как считаю, что предел unsinged int в 4 миллиона достаточен для работоспособности программы, так как для в практических целях мы используем время в гораздо меньших числах

В задаче 59 для всех переменных выбран вещественный тип double, так как все величины могут быть вещественными. Я не выбрал тип float, так как считаю, что точность в 7 знаков после запятой не приемлема в задачах, связанных с физикой. Я считаю, что в физике должна быть максимальная точность, поэтому мой выбор пал на тип double.