Лабораторная работа №1
Программирование алгоритмов линейной структуры

1. ЦЕЛЬ РАБОТЫ: приобретение практических навыков работы в интегрированной среде , изучение структуры программы на языке С.

2. ОСНОВНЫЕ СВЕДЕНИЯ

2.1. Алфавит и идентификаторы языка Си

Алфавит. В алфавит языка Си входят:
- прописные и строчные буквы латинского алфавита;
- цифры;
- специальные знаки “ , {} | [ ] ( ) + - / % \ ; ‘ . : ? < = > _ ! & * # ~ ^
- неизображаемые символы, например, пробел, табуляция.

Идентификатор. Последовательность букв, цифр, символов подчеркивания, начинающаяся с буквы или символа подчёркивания, считается идентификатором языка Си. Например, Lab_1, size22, _min, TIME, time.
Прописные и строчные буквы различаются, т. е. два последних идентификатора различны. Идентификаторы могут иметь различную длину, но компилятор учитывает не более 31-го символа.

2.2. Переменные

Переменная – это именованная область памяти. Отличительная особенность переменной – возможность связывать с её именем различные значения, совокупность которых определяется типом переменной. Каждая переменная перед её использованием в программе должна быть определена, т.е. для неё должна быть выделена память. Размер участка памяти зависит от типа переменной.

Основные типы переменных и их предельно допустимые значения:

char	-128…127	1 байт
unsigned char	0…255	1 байт
int	-32768…32767	2 байта
unsigned int	0…65535	2 байта
long	-2147483648…2147483647	4 байта
unsigned long	0…4294967295	4 байта
float	3.4E-38…3.4E+38	4 байта
double	1.7E-308…1.7E+308	8 байт
long double	3.4E-4932…1.1E+4932	10 байт

2.3. Операции

1) Операция присваивания : “=”.
2) Аддитивные операции : “ +”, ”-“;
3) Мультипликативные : “*”, ”/”, ”%”; % - получение остатка от деления, “/” — целочисленное деление (остаток отбрасывается);
4) Логические операции : && (and), || (or), ! (not);
5) Операции отношения : == (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно);
6) Операции модификации : *= присваивание после умножения (p *= 2; эквивалентно p=p*2;) /= присваивание после деления (p /= 2; эквивалентно p=p/2;) += присваивание после суммирования (a += b; эквивалентно a=a+b;)
-= присваивание после вычитания (a -= b; эквивалентно a=a-b;) %= присваивание после деления по модулю (n %= 3; эквивалентно n=n % 3;) a++ (--) постфиксное изменение (a++; эквивалентно a=a+1; значение a сначала используется, затем изменяется) ++(--)a префиксное изменение (++a; эквивалентно a=a+1; значение a сначала изменяется, затем используется).

2.4. Операторы ввода/вывода

1) **<u>Оператор вывода:</u>**

printf(<форматная строка>, переменные);

<u>Форматная строка</u> может включать текст для вывода на экран, спецификации преобразования данных, управляющие символы. В строке символов могут присутствовать спецификации преобразования данных, каждой из которой должен соответствовать аргумент. Если имеются спецификации преобразования, то вместо них выводятся значения соответствующих аргументов.

Для аргументов различных типов необходимо использовать соответствующие <u>спецификации преобразования</u>:

%d - для вывода целых чисел;
%c - для вывода образа символа, соответствующий аргумент должен содержать код символа;
%f - для вывода вещественного числа в виде целой и дробной части;
%e - для вывода вещественного числа в виде мантиссы и порядка;
%g - для вывода вещественного числа в виде %f или %e в зависимости от значения числа;
%u - для вывода беззнакового целого числа в десятичной системе счисления;

%o - для вывода беззнакового целого числа в восмеричной системе счисления;
%x - для вывода беззнакового целого числа в шестнадцатеричной системе счисления;
%s - для вывода на экран символьной строки, соответствующий аргумент должен быть адресом строки (т. е. именем символьного массива или строковой константой).

Для дополнительного управления преобразованием данных используются модификаторы преобразования. Они записываются между символом % и спецификацией преобразования.

%[флаг] [ширина] [. точность] [l][спецификатор типа преобразования]

[флаг] : “-“ или “+” ; ”-“ выравнивает текст вывода по левому краю; “+” выводит знак для положительных и отрицательных значений;
[ширина] : минимальный размер поля вывода;
[.точность] : для вещественных чисел количество знаков после десятичной точки;
[l] : для вывода целых чисел типа long;

Примеры спецификаций преобразований с модификаторами:

```
%-20.6lf , %6d, %8.4f
```

Пример:

```
double a=-78.98;

int c=24;

printf("a=%8.4lf, c=%6d", a, c);
```

На экране:

```
a=-78.9800, c=    24
```

Управляющие символы:
\n – перевод строки;
\t – табуляция;
\v – вертикальная табуляция;
\b – возврат на 1 символ;
\r – возврат на начало строки;
\a – звуковой сигнал.

2) **Оператор ввода:**

scanf(<форматная строка>, адреса переменных);

Функцию scanf рекомендуется использовать без лишних символов в формате, только спецификации преобразования, иначе может возникнуть непредсказуемая ситуация. Количество форматов определяется количеством переменных. Форматы при вводе переменных различных типов такие же, как и при выводе.

Обратите внимание, что в качестве аргументов передаются не значения переменных, а их адреса. Нужно сообщить функции адрес ячейки, в которую необходимо занести данные. Адреса переменных большинства типов обозначаются знаком &.

Пример:

```
int n;
scanf( "%d", &n );
```

Перед именами строк и массивов операция взятия адреса & не ставится, т. к. имя строки или массива само определяет их адрес.

Пример:

```
char name[41];
scanf( "%s", name );
```

2.5. Программирование алгоритма линейной структуры на языке Си

Чаще всего линейные алгоритмы используются для программирования вычислений по формулам. В этом случае удобно использовать набор стандартных функций Си, содержащихся в библиотеке и доступных при подключении заголовочного файла math.h :

sin (x) — sin x;
cos (x) — cos x;
tan(x) — tg x ;
log (x) — ln x;
log10(x) – lg x;
exp(x) — показательная функция e^x ;
sqrt (x) — корень квадратный от x;
pow(x, y) — x в степени y;
abs(x) — модуль x;
acos(x) — arccos x;
asin(x) – arcsin x;
atan(x) — arctg x;

sinh(x) — sh x;
cosh (x) — ch x;
tanh(x) — th x.

Для тригонометрических функций аргумент x измеряется в радианах и имеет тип double, как и значения функций.

Пример программы линейной структуры:

С клавиатуры вводятся значения длины гипотенузы c и угла x. Вычислить длины катетов и площадь прямоугольного треугольника.
Для вычислений воспользуемся формулами:

$$a = c \cdot \sin x;\ b = c \cdot \cos x;\quad S = ab/2;$$

Программа вычислений имеет вид:

```
 // вычисление длин сторон и площади треугольника:
#include <stdio.h>
#include <math.h>

int main(void)
{
    float a, b, c, x, S;
    printf ("\nEnter c:");
    scanf ("%f", &c);
    printf ("\nEnter x:");
    scanf ("%f", &x);
    a=c*sin(x);
    b=c*cos(x);
    S=a*b/2;
    printf("\na = %.2f     b = %.2f      S = %.2f ", a,
b, S);
    getchar();
    return 0;
}
```

<u>Объяснение примера:</u>

Комментарий в программе не влияет на компиляцию программы, а служит для разъяснения действий основных блоков текста и программы в целом. Однострочный комментарий действует от двух символов // до конца строки. Многострочный комментарий заключается в пары символов /* и */.
Первая строка программы #include <stdio.h> является **директивой препроцессора** для доступа к средствам ввода-вывода (связи с внешними устройствами), отсутствующими в самом языке Си. Вторая строка —

#include<math.h> является директивой компилятора для включения заголовочного файла math.h, обеспечивающего выполнение математических функций.

Программа на языке Си состоит из ряда функций, из которых **функция main (главная)** является обязательной и служит точкой входа в программу. В круглые скобки заключаются параметры функции, причем наличие круглых скобок обязательно, если даже список параметров пуст. В фигурные скобки заключаются составной **оператор** (несколько операторов).

Для придания тексту программы наглядности открывающая и соответствующая ей закрывающая фигурные скобки печатаются на одном уровне, а заключенный между ними текст сдвигается на 1-2 символа вправо, вложенный блок также сдвигается вправо и т.д. Образуется иерархия вложенных блоков, придающая программе на Си характерный вид.

В программе описаны **переменные** a, b, c, x, S вещественного типа (float). В отличие от других языков в Си учитывается регистр при определении имени переменной, т. е. s и S – разные переменные. **Функции** printf и scanf доступны при подключении заголовочного файла stdio.h и служат для вывода на экран и ввода с клавиатуры соответственно. Управляющие символы \n в функции printf служат для перевода на новую строку. Символ & в функции scanf указывает на адрес вводимой переменной. Ввод и вывод переменных вещественного типа производится в формате f. Признаком форматного вывода в функции printf является %. При выводе между знаком процента и форматной переменной f можно включить общую ширину поля вывода и число позиций после десятичной точки. **Оператор return** служит для выхода из функции main. Завершает текст программы закрывающая фигурная скобка, означающая конец функции main.

3 ЗАДАНИЕ

I. Наберите и выполните приведенную выше программу вычисления длин катетов и площади треугольника по значениям длины гипотенузы и острого угла, вводимым с клавиатуры.

Контрольные задания:

1. Добавить вывод значения cos(x).
2. Изменить строку в программе с выводом a, b, S так, чтобы значения a, b и S выводились каждое с новой строки.
3. Изменить строку программу так, чтобы числовые значения a, b, S были бы выровнены относительно точки.

Например:

```
a =      1.13
b =     21.53
S =     10.79
```

4. Изменить строку с выводом так, чтобы значения a, b, S выводились до третьего знака после точки.

5. Изменить программу так, чтобы в начале она спрашивала ваше имя и выводила затем строку "Hello, <ваше имя>!".

II. Составьте и выполните программы линейной структуры согласно следующим заданиям. Значения переменных не вводятся с клавиатуры, а задаются в самой программе. Полученное значение x выводится на экран.

Варианты:

1. $x=2y+3\ sh\ t- z$ при $y=2$; $t=5 / (1+y^2)$; $z=4$
2. $x=3\ y^2/ (4\ tg\ z-2t^2)$ при $t=0.5$; $z=6$; $y=t+2\ ctg\ z$.
3. $x=4y^2 /(4y\ e^z - 2t^3)$ при $t=1$; $z=3$; $y=sin\ t$.
4. $x=4\ ln\ y^3-z / t$ при $t=2$; $z=3$; $y=cos(t+z)$.
5. $x=6\ t^2- (ctg\ z+1)/ y^2$ при $y=2$; $z=4$; $t=sin(2+z)$.
6. $x=(8z^2+1)/(y\ e^t +t^2)$ при $z=1$; $t=2$; $y=tg\ t+z$.

Контрольные вопросы:

1. Понятие идентификатора.
2. Определение переменной. Основные типы переменных.
3. Типы операций.
4. Структура программы на языке Си.
5. Для чего используются заголовочные файлы?
6. Как откомпилировать программу?
7. Как выполнить программу?
8. Как просмотреть результат выполнения?