**Работа 2.О**

«Составление программ линейной структуры для вычисления значений математических выражений»

Использование стандартных библиотек математических функций.

Математические функции стандартной библиотеки С. Для реализации операций с основными типами данных предназначены операции языка программирования и функции стандартной библиотеки. Прототипы математических функций и определения некоторых констант содержатся в заголовочном файле math.h.

*Наиболее часто используемые функции:*

|  |  |
| --- | --- |
| *Функция* | *Краткое описание* |
| abs | нахождение абсолютного значения выражения типа int |
| acos | вычисление арккосинуса |
| asin | вычисление арксинуса |
| atan | вычисление арктангенса х |
| ceil | нахождение наименьшего целого, большего или равного х |
| cos | вычисление косинуса |
| exp | вычисление экспоненты |
| fabs | нахождение абсолютного значения типа double |
| floor | нахождение наибольшего целого, меньшего или равного х  кстати, floor(x+0.5) округлит x до ближайшего целого,  floor(x\*10n+0.5)/10n округлит x до заданного (n-ого) десятичного знака |
| labs | нахождение абсолютного значения типа long |
| log | вычисление натурального логарифма |
| log10 | вычисление логарифма по основанию 10 |
| modf | разложение х на дробную и целую часть |
| pow | вычисление х в степени у |
| sin | вычисление синуса |
| sqrt | нахождение квадратного корня |
| tan | вычисление тангенса |

Некоторые константы:

|  |  |
| --- | --- |
| M\_E | Основание натуральных логарифмов (e) |
| M\_PI | Отношение длины окружности к ее диаметру (π) |

Пример: Вычислить y=.

#include<math.h>

…

int main(){

…

y = pow(2, x)/cos(x)+pow(3, x);

…

}

Вычислить

l03_e60

#include <math.h>

int main()

{ double x; /\* параметр, который задаётся в программе \*/

double a=12.5, b=1.3; double c=14.1; double t2; /\* результат \*/

double ax; /\* рабочая переменная \*/

printf("Введите x >"); scanf("%lf ", &x);

ax=a\*x;

ax=sqrt(pow(c,2)-pow(b,2))\*tan(ax);

t2=log((ax+2)/(ax-2))/(2\*a\*b);

}

Задание 1. Вычислите значение выражения, используя математические функции стандартной библиотеки. Значения параметров вводятся с клавиатуры. Предусмотреть диалог с пользователем. Предполагается, что данные корректны (проверку выполнять не требуется).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *№* | *выражение* | *№* | *выражение* | *№* | *выражение* |
| 1 |  | 11 |  | 21 |  |
| 2 |  | 12 |  | 22 |  |
| 3 |  | 13 |  | 23 |  |
| 4 |  | 14 |  | 24 |  |
| 5 |  | 15 |  | 25 |  |
| 6 |  | 16 |  | 26 |  |
| 7 |  | 1 |  | 27 |  |
| 8 |  | 18 |  | 28 |  |
| 9 |  | 19 |  | 29 |  |
| 10 |  | 20 |  | 30 |  |

Дополнительное задание

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 16 |  |
|  |  | 17 |  |
|  |  | 18 |  |
|  |  | 19 |  |
|  |  | 20 |  |
|  |  | 21 |  |
|  |  | 22 |  |
|  |  | 23 |  |
|  |  | 24 |  |
|  |  | 25 |  |
|  |  | 26 |  |
|  |  | 27 |  |
|  |  | 28 |  |
|  |  | 29 |  |
|  |  | 30 |  |

**Работа 3.О**

«Составление простых программ разветвляющейся структуры»

Вычислить заданное целочисленное выражение для данных a, b.

Краткая теоретическая справка

Условный оператор if может использоваться в форме полной или неполной развилки.

| Неполная развилка | Полная развилка |
| --- | --- |
| if (Условие) {   БлокОпераций1; } | if (Условие) {   БлокОпераций1; } else {   БлокОпераций2; } |
|  |  |

В случае неполной развилки если **Условие** истинно, то **БлокОпераций1** выполняется, если **Условие** ложно, то **БлокОпераций1** не выполняется.  
В случае полной развилки если **Условие** истинно, то выполняется **БлокОпераций1**, иначе выполняется **БлокОпераций2**.

Оператор if может быть вложенным.

Пример :

#include <stdio.h>  
#include <stdlib.h> // для использования функции system  
int main() {  
  int key;  // объявляем целую переменную key  printf("Введите номер пункта, 1 или 2: ");  
  scanf("%d", &key);   // вводим значение переменной key  
  if (key == 1)        // если key = 1  
    printf("\n Выбран первый пункт"); // выводим сообщение  
  else if (key == 2)   // иначе если key = 2  
    printf("\n Выбран второй пункт"); // выводим сообщение  
  else                 // иначе  
    printf("\n Первый и второй пункты не выбраны"); // выводим сообщение  
  getchar(); getchar();  
  return 0;  
}

Если не использовать фигурные скобочки, то к телу оператора if будет относиться только один, первый оператор.

Пример :

#include <stdio.h>

#include <stdlib.h>

int main()

{    int age

    printf( "Сколько вам лет? " );       // спрашиваем пользователя о его возрасте

    scanf( "%d", &age );                 // ввод пользователем количества лет

    if ( age < 100 ) {                   // если введенный возраст меньше 100

        printf ("Вы очень молоды!\n" );  // программа сработала верно...

    }

    else if ( age == 100 ) {             // используем else для примера

        printf( "Молодость уже позади\n" ); // \n - символ перевода на новую строку.

    }

    else {

        printf( "Столько не живут\n" );  // если ни одно из выше-перечисленных условий не подошло, то программа покажет этот вариант ответа

    }

  return 0;

}

ЗАДАНИЕ 3.1.О

Задача 1. Вычислить заданное целочисленное выражение для данных a, b. Результат Х - тоже целочисленный. Исходные данные и результат должны проверяться на область допустимых значений.

|  |  |  |  |
| --- | --- | --- | --- |
| *№* | *Функция* | *№* | *Функция* |
| 1 |  | 11 |  |
| 2 |  | 12 |  |
| 3 |  | 13 |  |
| 4 |  | 14 |  |
| 5 |  | 15 |  |
| 6 |  | 16 |  |
| 7 |  | 17 |  |
| 8 |  | 18 |  |
| 9 |  | 19 |  |
| 10 |  | 20 |  |

**Работа 4**

**«**Составление программ разветвляющейся структуры (графика)**»**

## Краткая теоретическая справка

Для выполнения данной лабораторной работы потребуется вспомнить некоторые формулы и определения аналитической геометрии.

*Окружность* – геометрическое множество точек, равноудаленных от одной точки – центра окружности. Формула окружности выводится просто – опять по теореме Пифагора:

**A(x0, y0)**

**B(xi, yi)**

C(xi, y0)

xi

x0

y0

yi

x

y

0

r

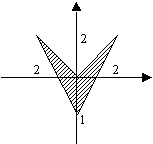


Т.е., квадрат расстояния от центра до точки на окружности равен квадрату радиуса окружности.

Для точек, лежащих внутри окружности, выполняется неравенство ; для точек, лежащих вне окружности – .

ХОД ВЫПОЛНЕНИЯ ТИПОВОГО ЗАДАНИЯ Т.4

Определить, попадает ли точка с координатами (x, y) в заштрихованную фигуру*.*



Алгоритм решения задачи:

1. Ввести значения координат x и y
2. Подставить значения координат в логическое выражение для проверки попадания точки в заштрихованную фигуру. Если логическое выражение истинно, вывести "да", иначе – "нет"

Основная сложность – определить логическое выражение.

Прежде всего, обратим внимание на то, что эту сложную фигуру целесообразно разбить на несколько более простых: треугольник, лежащий в I и IV координатных четвертях и треугольник, лежащий во II и III четвертях. Таким образом, точка может попасть внутрь одной из этих фигур, либо на линию, их ограничивающую. Количество отношений, описывающих какую-либо область, обычно совпадает с количеством линий, эту область ограничивающих. Чтобы точка попала внутрь области, необходима истинность каждого из отношений, поэтому над ними выполняется операция && (логическое И). Так как вся область была разбита на несколько, то между отношениями, описывающими каждую из них, используется операция || (логическое ИЛИ).

Уравнения прямых для первого (левого) треугольника (x≤0): y=-x, y=-1.5x-1; для второго треугольника (x≥0): y=x, y=1.5x-1. Полуплоскости задаются неравенствами, знак которых определяем простой подстановкой. Границы принадлежат фигуре, поэтому неравенства будут нестрогими.

Учитывая приведенные здесь соображения и записав уравнения всех ограничивающих фигуру линий, получаем логическое выражение:

(x <= 0) && (y >= –1.5 \* x – 1) && (y <= – x) ||

(x >= 0) && (y >= 1.5 \* x – 1) && (y <= x)

Определение переменных

Для реализации алгоритма нужны переменные для хранения значений координат x и y, оба числа – действительные числа. В условии не приведены требования к точности вычислений, рисунок представлен с довольно невысокой точностью, можно было бы использовать тип float. Но в соответствии с общим стилем программирования на C выберем для них тип double.

Текст программы:

void main(){

double x, y; //x, y – координаты точки

//Ввод координат точки:

printf("Введите координаты точки (x, y):\n");

scanf("%ld%ld", &x, &y);

//Проверка условия и получение результата:

if((x <= 0) && (y >= –1.5 \* x – 1) && (y <= – x) ||

(x >= 0) && (y >= 1.5 \* x – 1) && (y <= x))

printf("Точка (%.2f, %.2f) находится внутри фигуры\n", x, y);

else

printf("Точка (%.2f, %.2f) не попадает внутрь фигуры\n",x,y);

}

Тестовые примеры

(-0.5, -0.2) – точка в первом треугольнике

(0.5, 0) – точка во втором треугольнике

(0, 0) – точка на границе фигуры, на границе двух треугольников

(2, 1) – точка в I четверти, ниже фигуры

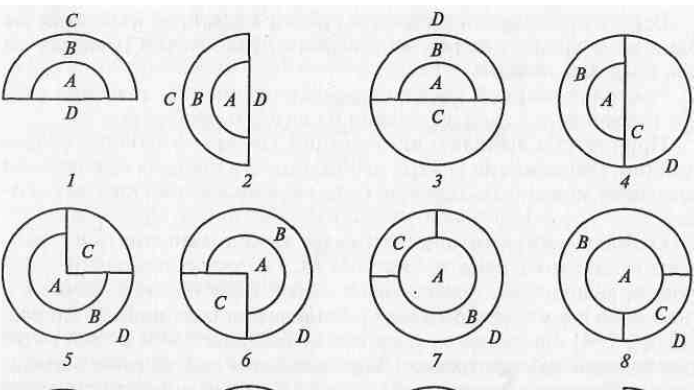
(1, 0.7) - точка в I четверти, выше фигуры

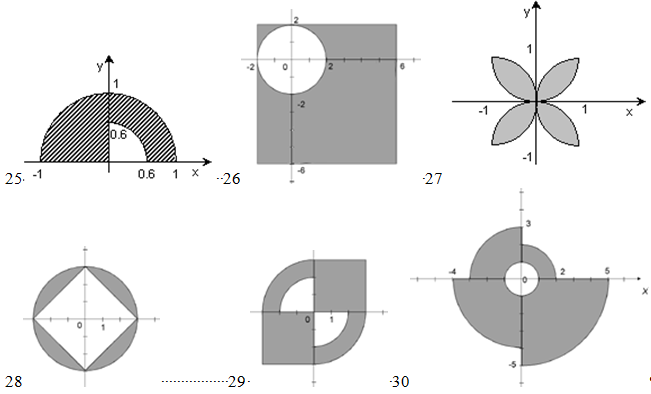
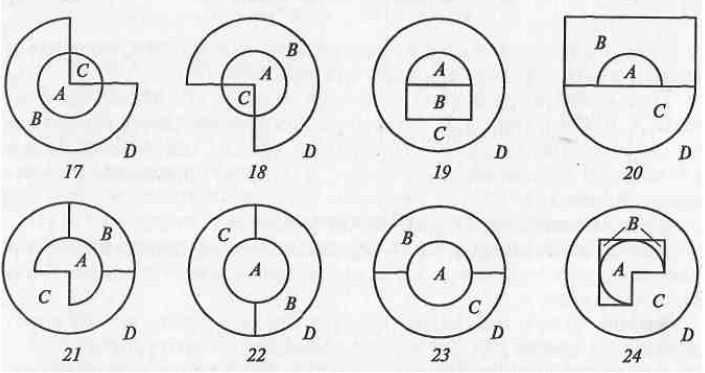
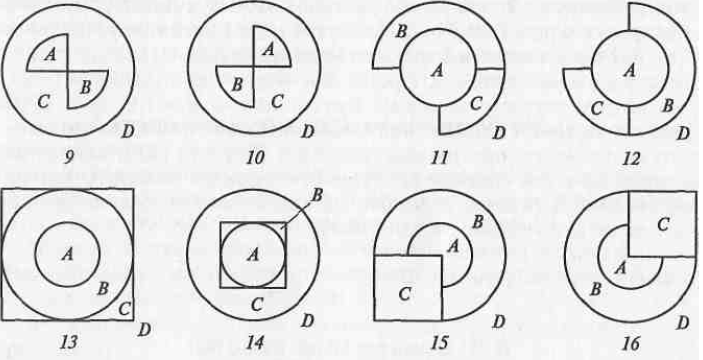
(-0.7, -0.7) – точка в III четверти, ниже фигуры

(-1, 0.7) - точка во II четверти, выше фигуры

ЗАДАНИЕ 4.1.О

. Определить, попадает ли точка с координатами (x, y) в заштрихованную фигуру, в области А, В, С, D.





**Работа 5**

«Составление программ разветвляющейся структуры с множественным выбором»

Краткая справочная информация

Оператор ветвления switch выполняется следующим образом:

* вычисляется значение **Выражение** в скобках оператора switch;
* полученное значение сравнивается с метками (**Константами**) во всех вариантах **case** , сравнение производится до тех пор, пока не будет найдена метка, соответствующая вычисленному значению целочисленного выражения;
* управление передаётся оператору, который соответствует значению выражения;
* если соответствующая метка не найдена, то выполнится **БлокОперацийПоУмолчанию**, описанный в опции default.

Как выражение, так и метки (константы) должны иметь значения целого или символьного типа

. Ключевое слово **case** вместе с *константой* служат просто *метками* и, если будут выполняться операторы для некоторого варианта **case**, то далее будут выполняться операторы всех последующих вариантов до тех пор, пока не встретится оператор **break**. Это позволяет связывать одну последовательность операторов с несколькими операторами **case**.

**break** – оператор завершения, прекращает выполнение ближайшего внешнего оператора **switch**, **while**, **do** или **for**.

Пример. Вывести название дня недели по его номеру***.***

#include <stdio.h>  
#include <stdlib.h>  
int main()  
{  int day;  
  system("chcp 1251");   system("cls");  
  printf("Введите номер дня недели: “); scanf(“%d“, &day);  
    
  switch (day)  
  {  
    case 1: printf("понедельник“); break;  
    case 2: printf("вторник“); break;  
    case 3: printf("среда“); break;  
    case 4: printf("четверг“); break;  
    case 5: printf("пятница“); break;  
    case 6: printf("суббота“); break;  
    case 7: printf("воскресенье“); break;  
    default: printf("Неверно введен день недели“); break;  
  }  
   return 0;  
}

ЗАДАНИЕ 5.1

Решить задачу с использованием оператора switch

|  |  |
| --- | --- |
| *№* | *Формулировка задачи* |
| 1 | Дан номер месяца (1 – январь, 2 – февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.). |
| 2 | Дан номер месяца. Определить название месяца по его номеру, например, 1 - январь, 2 - февраль и т. д. |
| 3 | Дан номер месяца (1 – январь, 2 – февраль, ...). Вывести число дней в этом месяце для невисокосного года. |
| 4 | Установить название дня недели по его номеру, например, 1 - понедельник, 2 - вторник и т.д. |
| 5 | Определить название заданной цифры, например 1 - один, 5 - пять т.д. |
| 6 | Дано целое число в диапазоне 0 – 9. Вывести строку – название соответствующей цифры на русском языке (0 – "ноль", 1 – "один", 2 – "два", ...). |
| 7 | Дано целое число в диапазоне 1 – 5. Вывести строку – словесное описание соответствующей оценки (1 – "плохо", 2 – "неудовлетворительно", 3 – "удовлетворительно", 4 – "хорошо", 5 – "отлично"). |
| 8 | Над числами определены 4 арифметические действия: + – сложение, - – вычитание, \* – умножение, / – деление. Заданы действие и два числа A и B (В не равно нулю) в формате: «А» «действие» «В». Выполнить над числами указанное действие и вывести результат. |
| 9 | Единицы длины пронумерованы следующим образом: 1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах. |
| 10 | Единицы массы пронумерованы следующим образом: 1 – килограмм, 2 – миллиграмм, 3 – грамм, 4 – тонна, 5 – центнер. Дан номер единицы массы и масса тела M в этих единицах (вещественное число). Вывести массу данного тела в килограммах. |
| 11 | Робот может перемещаться в четырех направлениях ("С" – север, "З" – запад, "Ю" – юг, "В" – восток) и принимать три цифровые команды: 0 – продолжать движение, 1 – поворот налево, –1 – поворот направо. Дан символ C – исходное направление робота и число N – посланная ему команда. Вывести направление робота после выполнения полученной команды. |
| 12 | Локатор ориентирован на одну из сторон света ("С" – север, "З" – запад, "Ю" – юг, "В" – восток) и может принимать три цифровые команды: 1 – поворот налево, –1 – поворот направо, 2 – поворот на 180 градусов. Дан символ C – исходная ориентация локатора и числа N1 и N2 – две посланные ему команды. Вывести ориентацию локатора после выполнения данных команд. |
| 13 | Элементы окружности пронумерованы следующим образом: 1 – радиус (R), 2 – диаметр (D), 3 – длина (L), 4 – площадь круга (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке). В качестве значения Pi использовать 3.14. |
| 14 | Элементы равнобедренного прямоугольного треугольника пронумерованы следующим образом: 1 – катет (a), 2 – гипотенуза (c), 3 – высота, опущенная на гипотенузу (h), 4 – площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке). |
| 15 | Элементы равностороннего треугольника пронумерованы следующим образом: 1 – сторона (a), 2 – радиус вписанной окружности (R1), 3 – радиус описанной окружности (R2), 4 – площадь (S). Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке). |
| 16 | Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, предшествующей указанной. |
| 17 | Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, следующей за указанной. |
| 18 | Дано целое число в диапазоне 20 – 69, определяющее возраст (в годах). Вывести строку – словесное описание указанного возраста, обеспечив правильное согласование числа со словом "год", например: 20 – "двадцать лет", 32 – "тридцать два года", 41 – "сорок один год". |
| 19 | Дано целое число в диапазоне 100 – 999. Вывести строку – словесное описание данного числа, например: 256 – "двести пятьдесят шесть", 814 – "восемьсот четырнадцать". |
| 20 | В восточном календаре принят 60-летний цикл, состоящий из 12-летних подциклов, обозначаемых названиями цвета: зеленый, красный, желтый, белый и черный. В каждом подцикле годы носят названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. По номеру года вывести его название, если 1984 год был началом цикла – годом зеленой крысы. |