Лабораторная работа №5. Процедуры и функции.

Цель л/р: рассмотрение возможных методов реализации алгоритмов с использованием функций и процедур средствами Lazarus и Free Pascal.

Оборудование: Lazarus, Draw.io, Word.

Задание 1.

1. Перевести дюймы в сантиметры от 0 до 100 дюймов. (1 дюйм=2.5 см). Результаты вывести в виде таблицы. Операторы для формирования вывода таблицы оформить в виде пользовательской процедуры.
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| q | Количество дюймов для расчёта | byte |
| i | Счётчик цикла для многократного обращения к процедуре | byte |
| r | Входной аргумент процедуры output | byte |
| output | Процедура вывода значений дюймов и соответствующих значений сантиметров | procedure |

1. Код программы:

program first;

var i, q:byte;

procedure output(r);

begin

writeln('Inches Centimeters');

write(' ');

write(r);

write(' ');

writeln(r\*2.5:5:1);

end;

begin

q:= 100;

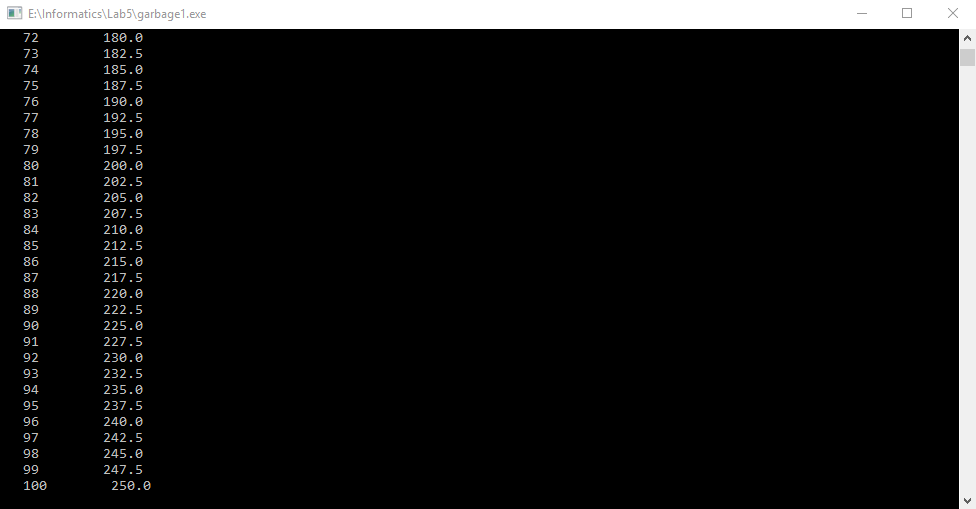
for i:= 0 to q do

output(r);

readln();

end.

1. Результаты выполненной работы:

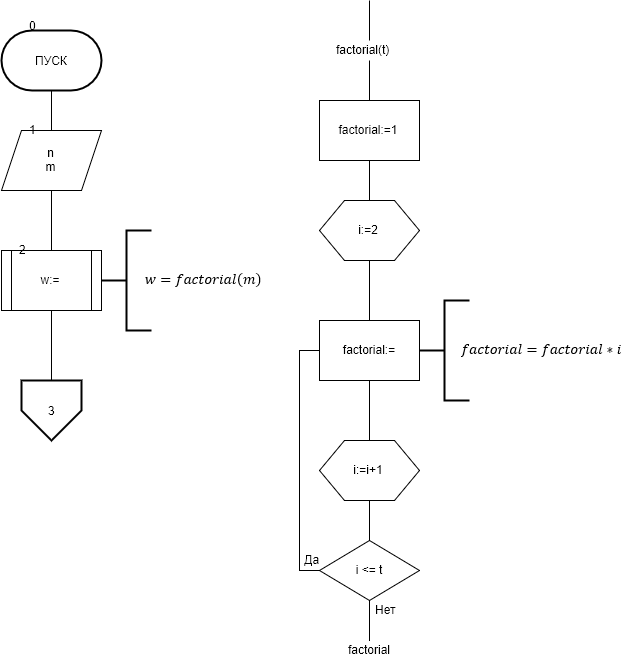


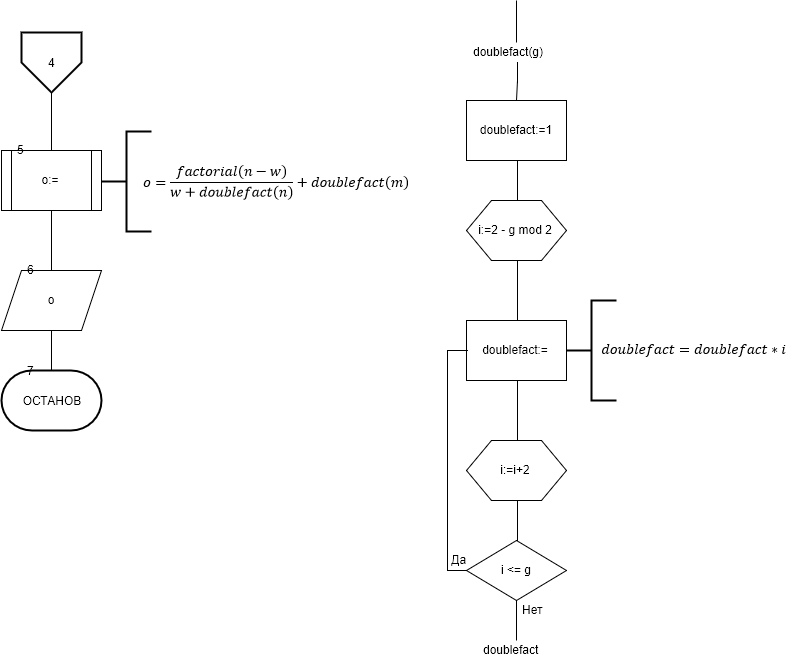
1. Анализ результатов вычисления:

Программа последовательно выводит значения дюймов от 0 до q и напротив значения сантиметров от 0 до q\*2,5. Использование процедуры позволяет обойтись без выделения лишней переменной для хранения значений сантиметров.

Задание 2.



1. Напишите программу
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| n | Заданное значение | word |
| m | Заданное значение | word |
| w | Переменная, хранящее значение факториала m, используется дважды в конечном выражении | longword |
| t | Входной аргумент функции factorial | word |
| i | Счётчик циклов в обеих функциях | byte |
| factorial | Выходное значение функции factorial, значение факториала от данного числа | longword |
| g | Входной аргумент функции doublefact | word |
| doublefact | Выходное значение функции doublefact, значение двойного факториала от заданного числа | longword |
| o | Конечный результат | real |

1. Код программы:

program factorials;

var n, m:word;

w:longword;

o:real;

function factorial(t:word):longword;

var i:byte;

begin

factorial:=1;

for i:=2 to t do

factorial:=factorial\*i;

end;

function doublefact(g:word):longword;

var i:byte;

begin

doublefact:=1;

i:=2-g mod 2;

while i <= g do begin

doublefact:=doublefact\*i;

i:=i+2;

end;

end;

begin

writeln('Enter n');

readln(n);

writeln('Enter m');

readln(m);

w:=factorial(m);

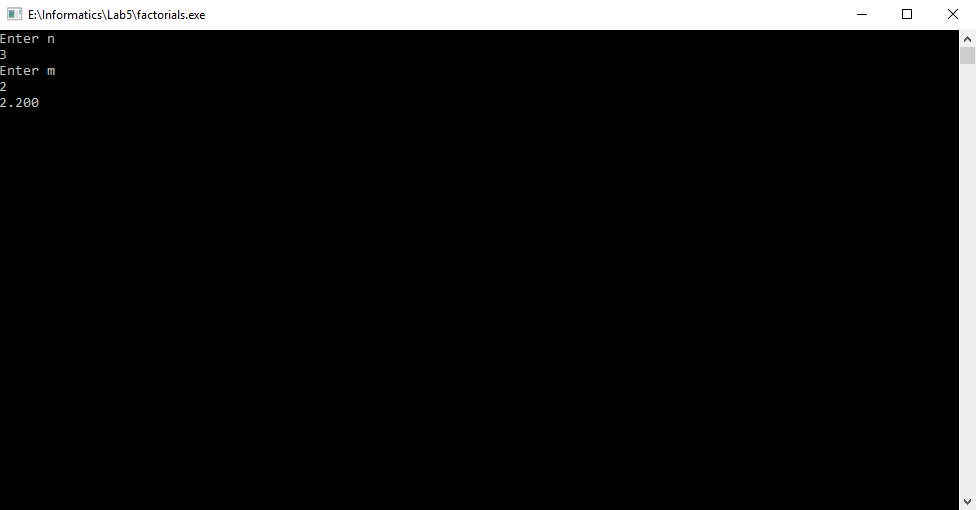
o:=factorial(n-w)/(w+doublefact(n))+ doublefact(m);

writeln(o:3:3);

readln();

end.

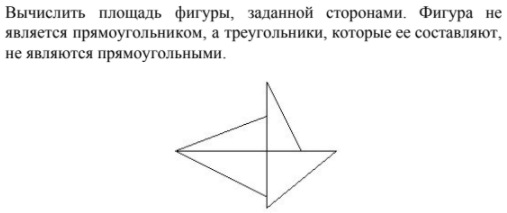
1. Результаты выполненной работы:



2

1. Анализ результатов вычисления:

Вычисляемое выражение имеет смысл при n>=m!, т.к. в противном случае в числителе окажется факториал от отрицательного числа. Кроме того, вычисления факториалов и их сумм ограничены 32 битами памяти, выделяемыми под тип longword. Функция factorial предназначена для вычисления факториала данного числа. Функция doublefact вычисляет двойной факториал, равный 1\*3\*5\*…\*n при нечётном n и 2\*4\*6\*…\*n при чётном n. Перед подсчётом финального выражения вычисляется факториал m ввиду его повторения.

Задание 3.

1. Математическая модель:
2. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| num | Количество треугольников | byte |
| S | Конечный результат, сумма площадей фигур | real |
| i | Счётчик цикла | byte |
| a | Первая вводимая сторона треугольника | real |
| b | Вторая вводимая сторона треугольника | real |
| c | Третья вводимая сторона треугольника | real |
| m | Аргумент функции trS, принимает значение одной из сторон треугольника | real |
| n | Аргумент функции trS, принимает значение одной из сторон треугольника | real |
| l | Аргумент функции trS, принимает значение одной из сторон треугольника | real |
| p | Полупериметр треугольника | real |
| trS | Площадь данного треугольника | real |

1. Код программы:

program triangles;

var i:byte;

a,b,c,num,S:real;

function trS(m,n,l:real):real;

var p:real;

begin

p:=(m+n+l)/2;

trS:= exp(0.5\*ln(p\*(p-m)\*(p-n)\*(p-l)));

end;

begin

num:=4;

S:=0;

for i:=1 to num do begin

readln(a);

readln(b);

readln(c);

S:=S+trS(a,b,c);

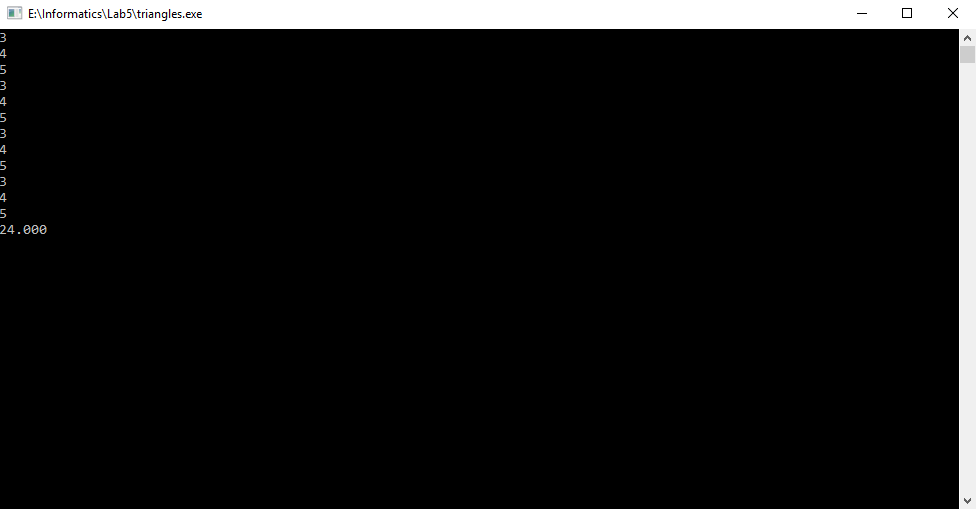
end;

writeln(S:5:3);

readln();

end.

1. Результаты выполненной работы:



1. Анализ результатов вычисления:

Программа состоит из цикла, в котором на каждом шаге вводятся величины сторон треугольника, после чего осуществляется вызов функции, вычисляющей последовательно полупериметр, затем площадь данного треугольника по формуле Герона. Цикл расчитан на 4 итерации согласно условию. После выполнения цикла на экран выводится значение суммы треугольников.

Задание 4.

1. С клавиатуры вводится число. Вывести на экран столько элементов ряда Фибоначчи, сколько указал пользователь. Вычисление ряда организовать в функцию. Например, если на ввод поступило число 6, то вывод должен содержать шесть первых чисел ряда Фибоначчи: 1 2 3 5 8 13.
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| n | Вводимое кол-во чисел | byte |
| q | Счётчик цикла | byte |
| fib | Выходное значение функции fib, выводит число Фибоначчи с индексом count | word |
| f1 | Число, предшествующее вычисляемому числу Фибоначчи. | word |
| f2 | Число, предшествующее вычисляемому числу Фибоначчи на две позиции. | word |
| i | Счётчик цикла, используемого в функции fib | byte |
| count | Аргумент функции fib | byte |

1. Код программы:

program fibonacci;

var n,q:byte;

function fib(count:byte):word;

var f1,f2:word;

i:byte;

begin

f1:=0;

f2:=1;

for i:=1 to count do begin

fib:=f1+f2;

f1:=f2;

f2:=fib;

end;

end;

begin

writeln('Enter Fibonacci numbers quantity');

readln(n);

for q:=1 to n do

writeln(fib(q));

readln();

end.

1. Результаты выполненной работы:
2. Анализ результатов вычисления:

Предельным вычисляемым значением является 24 число Фибоначчи 46368, далее происходит переполнение переменной word. Функция fib используется для вычисления числа Фибоначчи с индексом count. Цикл, расположенный в теле программы, позволяет вывести на экран заданное кол-во чисел.

Вывод:

Процедуры и функции особенно эффективны при необходимости в многоповторных расчётах по единому алгоритму. Кроме того, их применение упрощает читаемость кода, позволяет выделить блоки, ответственные за определенные функции.