Лабораторная работа №3

Условие:

1. {(x - 2)^2 + 6, 1 < x < 2  
   y = {  
    {ln(x + 3 \* sqrt(x)), x => 2;
2. Написать программу, которая определяет: кратна ли трем сумма цифр двухзначного числа;
3. Дана точка на плоскости с координатами(х, у). Составить программу, которая выдает одно из сообщений "Да", "Нет", "На границе" в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе;
4. Составить программу(при решении данных задач использовать оператор switch или вложенные операторы if): с некоторой даты по настоящий день прошло n дней, найти неизвестную дату;
5. Составить программу решения задачи: в соревнованиях по фигурному катанию N судей независимо выставляют оценки спортсмену. Затем из объявленных оценок удаляют самую высокую (одну, если самую высокую оценку выставили несколько судей). Аналогично поступают с самой низкой оценкой. Для оставшихся оценок вычисляется среднее арифметическое, которое и становится зачетной оценкой. По заданным оценкам судей определить зачетную оценку спортсмена;
6. Составить программу решения задачи: по дате (месяц и день) рождения определите знак Зодиака;
7. Даны цифры двух целых чисел: двузначного а2а1 и однозначного b, где а1 – число единиц, а2 – число десятков. Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное). Слагаемое – двузначное число и число-результат не определить; условный оператор не использовать;
8. Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for): только положительные целые числа из диапазона от А до В;
9. Вывести на экран числа следующим образом:  
   1  
   1 2  
   1 2 3  
   1 2 3 4  
   1 2 3 4 5
10. Решите задачу: дано натуральное число n. Получить все натуральные числа, меньшие n и взаимно простые с ним (два натуральных числа называются взаимно простыми, если их наибольший общий делитель равен 1).
11. Дано натуральное число. Определить: сколько раз в нем встречаются цифры х и у;
12. Решите задачу: известны оценки каждого из студентов университета по физике. Посчитать количество пятерок, количество четверок, количество троек и количество двоек.

Решение:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp9

{

class Program

{

public static int NOD(int a, int b)

{

if (a == b)

return a;

else

if (a > b)

return NOD(a - b, b);

else

return NOD(b - a, a);

}

static void Main(string[] args)

{

/\*

Задание 1

{(x - 2)^2 + 6, 1 < x < 2

y = {

{ln(x + 3 \* sqrt(x)), x >= 2

\*/

Console.WriteLine("Задание 1");

start1: Console.Write("Введите х для промежутка (1; +∞): ");

double y1 = 0, x1 = 0;

x1 = double.Parse(Console.ReadLine());

if (x1 >= 2)

{

y1 = Math.Log10(x1 + 3 \* Math.Sqrt(x1));

}

else if (x1 > 1 && x1 < 2)

{

y1 = Math.Pow((x1 - 2), 2) + 6;

}

else

{

goto start1;

}

Console.WriteLine("y1 = {0}", y1);

Console.WriteLine();

/\*

Задание 2

Написать программу, которая определяет: кратна ли трем сумма цифр двухзначного числа.

\*/

Console.WriteLine("Задание 2");

Console.WriteLine("Введите двухзначное число");

string two = Console.ReadLine();

double buf = two[0] + two[1] - 96;

Console.WriteLine("Сумма цифр = {0}", buf);

if (buf % 3 == 0)

{

Console.WriteLine("Сумма цифр кратна трем");

}

else

{

Console.WriteLine("Сумма цифр не кратна трем");

}

Console.WriteLine();

/\*

Задание 3

Дана точка на плоскости с координатами(х, у).

Составить программу, которая выдает одно из сообщений "Да", "Нет", "На границе"

в зависимости от того, лежит ли точка внутри заштрихованной области,

вне заштрихованной области или на ее границе.

\*/

Console.WriteLine("Задание 3");

Console.Write("x=");

double x3, y;

x3 = double.Parse(Console.ReadLine());

Console.Write("y=");

y = double.Parse(Console.ReadLine());

if (x3 < 50 && x3 > -50 && y < 25 && y > -25)

Console.WriteLine("Да");

else if (((x3 == 50 || x3 == -50) && (y <= 25 && y >= -25)) || ((x3 <= 50 && x3 >= -50) && (y == 25 || y == -25)))

Console.WriteLine("На границе");

else Console.WriteLine("Нет");

Console.WriteLine();

/\*

Задание 4

Составить программу(при решении данных задач использовать оператор switch или вложенные операторы if):

С некоторой даты по настоящий день прошло n дней, найти неизвестную дату.

\*/

int n, day4, month4, year4, all, day4new = 0, month4new = 0, year4new = 0;

Console.WriteLine("Задание 4");

Console.Write("Введите количество пройденных дней: n = ");

n = Int32.Parse(Console.ReadLine());

DateTime dateGregorian = DateTime.Today;

day4 = dateGregorian.Day;

month4 = dateGregorian.Month;

year4 = dateGregorian.Year;

all = day4 + month4 \* 30 + year4 \* 365 - n;

if (all > 365)

{

year4new = all / 365;

all = all - year4new \* 365;

}

if (all > 30)

{

month4new = all / 30;

day4new = all - month4new \* 30;

}

if (day4new == 0)

{

day4new = 30;

month4new = month4new - 1;

}

Console.WriteLine("Полученная дата: {0:00}.{1:00}.{2:0000}", day4new, month4new, year4new);

Console.WriteLine();

/\*

Задание 5

Составить программу решения задачи:

В соревнованиях по фигурному катанию N судей независимо выставляют оценки спортсмену.

Затем из объявленных оценок удаляют самую высокую (одну, если самую высокую оценку выставили несколько судей).

Аналогично поступают с самой низкой оценкой. Для оставшихся оценок вычисляется среднее арифметическое,

которое и становится зачетной оценкой. По заданным оценкам судей определить зачетную оценку спортсмена.

\*/

Console.WriteLine("Задание 5");

int bufMax, bufMin;

double srednee;

Console.Write("Введите кол-во судей: ");

int N = Int32.Parse(Console.ReadLine());

int[] Ocenki = new int[N];

Random rand = new Random();

srednee = 0;

for (int i = 0; i < N; i++) //Задаем случайные оценки и выводим на экран, подсчитываем общую сумму оценок

{

Ocenki[i] = rand.Next(0, 6);

Console.Write("{0} ", Ocenki[i]);

srednee += Ocenki[i];

}

int maxValue = Ocenki.Max<int>(); //Находим мин и макс

int minValue = Ocenki.Min<int>();

Console.WriteLine();

bufMax = 0;

bufMin = 0;

for (int i = 0; i < N; i++)

{

if (maxValue == Ocenki[i])

bufMax += 1;

if (minValue == Ocenki[i])

bufMin += 1;

}

if (bufMax > 1)

{

srednee = srednee - maxValue \* bufMax;

N = N - bufMax;

}

if (bufMin > 1)

{

srednee = srednee - minValue \* bufMin;

N = N - bufMin;

}

srednee = srednee / N;

Console.WriteLine("Зачетная оценка: {0}", srednee);

Console.WriteLine();

/\*

Задание 6

Составить программу решения задачи:

По дате (месяц и день) рождения определите знак Зодиака.

\*/

string month, day, zodiak = "";

Console.WriteLine("Задание 6");

Console.Write("Введите месяц: ");

month = Console.ReadLine();

Console.Write("Введите день: ");

day = Console.ReadLine();

Console.WriteLine();

if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 3) | (Convert.ToInt32(day) < 21 && Convert.ToInt32(month) == 4))

{

zodiak = "Овен";

}

else if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 4) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 5))

{

zodiak = "Телец";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 5) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 6))

{

zodiak = "Близнецы";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 6) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 7))

{

zodiak = "Рак";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 7) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 8))

{

zodiak = "Лев";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 8) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 9))

{

zodiak = "Дева";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 9) | (Convert.ToInt32(day) < 24 && Convert.ToInt32(month) == 10))

{

zodiak = "Весы";

}

else if ((Convert.ToInt32(day) > 23 && Convert.ToInt32(month) == 10) | (Convert.ToInt32(day) < 23 && Convert.ToInt32(month) == 11))

{

zodiak = "Скорпион";

}

else if ((Convert.ToInt32(day) > 22 && Convert.ToInt32(month) == 11) | (Convert.ToInt32(day) < 22 && Convert.ToInt32(month) == 12))

{

zodiak = "Стрелец";

}

else if ((Convert.ToInt32(day) > 21 && Convert.ToInt32(month) == 12) | (Convert.ToInt32(day) < 21 && Convert.ToInt32(month) == 1))

{

zodiak = "Козерог";

}

else if ((Convert.ToInt32(day) > 20 && Convert.ToInt32(month) == 1) | (Convert.ToInt32(day) < 20 && Convert.ToInt32(month) == 2))

{

zodiak = "Водолей";

}

else

{

zodiak = "Рыба";

}

Console.WriteLine("Ваш знак задиака: {0}", zodiak);

Console.WriteLine();

/\*

Задание 7

Даны цифры двух целых чисел: двузначного а2а1 и однозначного b, где а1 – число единиц, а2 – число десятков.

Получить цифры числа, равного сумме заданных чисел (известно, что это число двузначное).

Слагаемое – двузначное число и число-результат не определить; условный оператор не использовать.

\*/

Console.WriteLine("Задание 7");

int a1, a2, b, c1, c2;

Console.Write("Введите a2: ");

a2 = Int32.Parse(Console.ReadLine());

Console.Write("Введите a1: ");

a1 = Int32.Parse(Console.ReadLine());

Console.Write("Введите b: ");

b = Int32.Parse(Console.ReadLine());

if (a1 + b >= 10)

{

c1 = a1 + b - 10;

c2 = a2 + 1;

}

else

{

c1 = a1 + b;

c2 = a2;

}

Console.WriteLine("Цифры полученного числа: {0}, {1}", c2, c1);

Console.WriteLine();

/\*

Задание 8

Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for):

Только положительные целые числа из диапазона от А до В ( );

\*/

Console.WriteLine("Задание 8");

Console.Write("Задайте диапазон от A: ");

int A = Int32.Parse(Console.ReadLine());

Console.Write(" до B: ");

int B = Int32.Parse(Console.ReadLine());

int temp = A;

for (int i = A; i < B; i++)

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

Console.WriteLine();

temp = A;

while (temp < B)

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

Console.WriteLine();

temp = A;

do

{

if (temp > 0)

{

Console.Write("{0} ", temp);

}

temp += 1;

}

while (temp < B);

Console.WriteLine();

/\*

Задание 9

Вывести на экран числа следующим образом:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

\*/

Console.WriteLine("Задание 9");

for (int i = 1; i < 7; i++)

{

for (int j = 1; j < i; j++)

{

Console.Write("{0} ", j);

}

Console.WriteLine();

}

Console.WriteLine();

/\*

Задание 10

Решите задачу:

Дано натуральное число n.

Получить все натуральные числа, меньшие n и взаимно простые с ним

(два натуральных числа называются взаимно простыми, если их наибольший общий делитель равен 1).

\*/

Console.WriteLine("Задание 10");

Console.Write("Введите n: ");

int n10 = Int32.Parse(Console.ReadLine());

HashSet<int> primes = new HashSet<int>(Enumerable.Range(2, n10 - 1));

for (int i10 = 2; i10 < n10; i10++)

{

if (primes.Contains(i10))

{

primes.RemoveWhere(x => x > i10 && x % i10 == 0);

}

}

primes.Remove(2);

Console.Write("1, ");

Console.WriteLine(String.Join(", ", primes));

Console.WriteLine();

/\*

Задание 11

Дано натуральное число. Определить:

Сколько раз в нем встречаются цифры х и у;

\*/

Console.WriteLine("Задание 11");

string nat, x11, y11;

Console.Write("Введите натуральное число: ");

nat = Console.ReadLine();

Console.Write("Введите x: ");

x11 = Console.ReadLine();

Console.Write("Введите y: ");

y11 = Console.ReadLine();

int count;

count = (nat.Length - nat.Replace(x11, "").Length) / x11.Length;

count = count + (nat.Length - nat.Replace(y11, "").Length) / y11.Length;

Console.WriteLine("Символы встречаются: {0}", count);

Console.WriteLine();

/\*

Задание 12

Решите задачу:

Известны оценки каждого из студентов университета по физике.

Посчитать количество пятерок, количество четверок, количество троек и количество двоек.

\*/

Console.WriteLine("Задание 12");

Console.Write("Введите количество студентов: ");

int students = Int32.Parse(Console.ReadLine());

int[] Students = new int[students];

Random rand12 = new Random();

int o5 = 0, o4 = 0, o3 = 0, o2 = 0;

for (int i12 = 0; i12 < students; i12++)

{

Students[i12] = rand12.Next(0, 6);

Console.Write("{0} ", Students[i12]);

if (Students[i12] == 5)

{

o5++;

}

if (Students[i12] == 4)

{

o4++;

}

if (Students[i12] == 3)

{

o3++;

}

if (Students[i12] == 2)

{

o2++;

}

}

Console.WriteLine();

Console.WriteLine("Количество 5 = {0}, 4 = {1}, 3 = {2}, 2 = {3}", o5, o4, o3, o2);

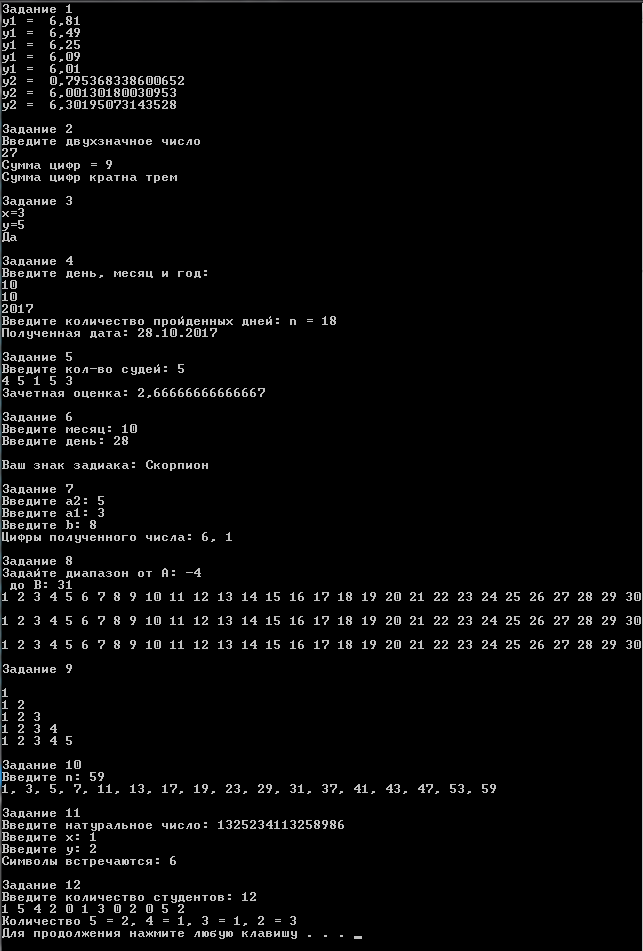
Console.ReadKey();

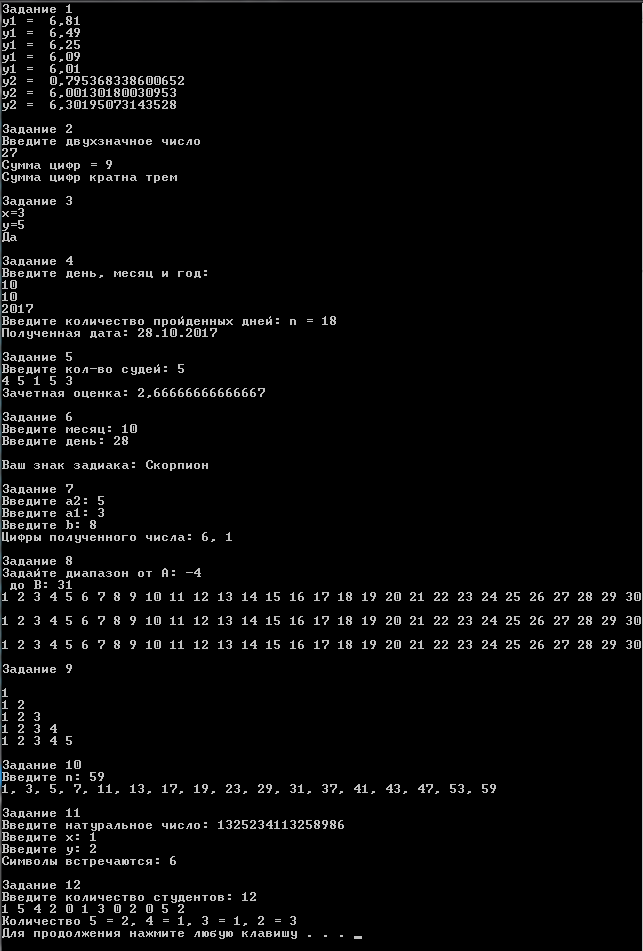
}

}

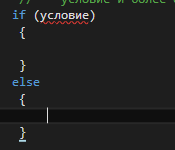
}

Тестирование:





Контрольные вопросы:

1. Оператор return, goto, break, continue;
2. Имеют условный оператор, структурно аналогичный Паскалю. Отличие состоит в том, что условие должно быть записано в круглых скобках, а вместо ключевых слов begin и end используются фигурные скобки {}.
3. Метка (англ. label) — символьное имя, идентификатор для более удобного указания данных и кода в языках программирования. Позволяет программисту обходиться без вычисления и пересчёта адресов и смещений внутри программы (эти действия за него выполняет компилятор).
4. Выход из вложенных циклов, Обработка ошибок, Автогенерация кода.
5. 
6. Оператор безусловного перехода Оператор goto переходит при выполнении программы к определенному оператору программы, перед которым находится метка.
7. Оператор switch сравнивает значение одной переменной с несколькими константами. Основной формат для использования оператора множественного выбора switch case
8. switch - оператор множественного выбора, может обрабатывать различные варианты, в то время как if лишь один. Фактически, switch выглядит как множество if else if else ... А работают они с выражениями, а не типами данных. От последних вобще ничего не зависит.
9. Они оба условные.
10. Применится дефолтное значение.
11. Когда имеются сомнения в написанном коде.
12. Для выхода из цикла.
13. Значение условия выполнения цикла должно быть определено до начала цикла; если значение условия истинно, то выполняется тело цикла, после чего повторяется проверка условия. Если условие ложно, то происходит выход из цикла; хотя бы один из операторов, входящих в тело цикла, должен влиять на значение условия выполнения цикла, иначе цикл будет повторяться бесконечное число раз.
14. Существует два основных вида области видимости: локальная область видимости и глобальная область видимости. Переменная, объявленная вне всех функций, помещается в глобальную область видимости. Доступ к таким переменным может осуществляться из любого места программы. Такие переменные располагаются в глобальном пуле памяти, поэтому время их жизни совпадает со временем жизни программы.
15. Переменная, объявленная внутри блока (часть кода, заключенная в фигурные скобки), принадлежит локальной области видимости. Такая переменная не видна (поэтому и недоступна) за пределами блока, в котором она объявлена. Самый распространенный случай локального объявления – переменная, объявленная внутри функции. Переменная, объявленная локально, располагается на стеке, и время жизни такой переменной совпадает со временем жизни функции.Так как областью видимости локальной переменной является блок, в котором она объявлена, то существует возможность объявлять переменные с именем, совпадающим с именами переменных, объявленных в других блоках; а также объявленных на более верхних уровнях, вплоть до глобального.
16. Верно.
17. В конце.
18. Верно.
19. Верно. Параметр задаётся заранее.
20. Верное утверждение.
21. По своей структуре, вложенные циклы напоминают мне коробки разных размеров, как на нашем рисунке. Эти коробки легко можно сложить друг в друга и поместить в одну большую коробку. Так и с циклами. Программе, чтобы добраться к вложенному циклу, необходимо сначала приступить к выполнению внешнего цикла (открыть крышку большой коробки), выполнить, то что прописано в коде до вложенного цикла и потом уже приступить к выполнению этого цикла.
22. После завершение одного цикла.
23. Оптимизация циклов представляет – важный фактор в повышении производительности приложений с параллельной обработкой данных. Такого рода оптимизации, как объединение, перестановка и развертывание циклов, обычно используются для улучшения степени разбиения, баланса нагрузки и эффективной локализации данных, и при этом влекут за собой незначительные затраты связанные с синхронизацией и другими издержками параллелизма. Основное правило - для распараллеливания лучше всего подходят циклы с большим числом итераций. Большое число итераций позволяет улучшить баланс нагрузки благодаря наличию большего числа задач для распределения между потоками. Также следует учитывать объем работы, выполняемый за одну итерацию. Если в статье отдельно не оговаривается иное, мы подразумеваем, что объем вычислений в каждой итерации цикла (приблизительно) одинаков для всех итераций данного цикла.
24. Любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).
25. Оператор цикла foreach в Си-шарп служит для перебора элементов коллекции. К коллекциям относятся массивы, списки List и пользовательские классы коллекций. В данном операторе не нужно создавать переменную-счетчик для доступа к элементам коллекции, в отличии от других циклов. Оператор foreach имеет следующую структуру:

Foreach ([тип] [переменная] in [коллекция])

{

//тело цикла

}