1. Вычислите значения констант E и Pi с 24 значащими цифрами с помощью команды N. Представьте в виде десятичных дробей числа Pi^(-20) и Pi^(-21). Что произойдет, если к этим числам применить команду Chop? (заменяет приблизительные вещественные числа в выражении, близкие к нулю, точным целым числом 0.)

$$N[E, 24]$$

```
N[Pi, 24]

Pi^(-20) // N

Pi^(-21) // N

Chop[Pi^(-20) // N]

Chop[Pi^(-21) // N]
```

2. С помощью команды Solve (см. Help) решите уравнение x^7 + x^3 - 3 = 0 ``в радикалах и приближенно. Используйте команды Equal, N и ReplaceAll. С помощью команды Plot нарисуйте график функции y = x^7 + x^3 - 3 и проверьте предыдущий ответ. Как проверить комплексные корни?

Замечание. Результатом работы команды Solve является подстановка (x -> 3), а не присвоение (x = 3). Это связано с тем, что повторный прогон программы без перезагрузки будет невозможен.

```
roots=Solve[x^7+x^3-3==0,x]
roots = N[roots]
x/.roots
x^7+x^3-3/.roots[[7]]
Plot[x^7+x^3-3, {x, -2, 2}]
ComplexPlot3D[x^7+x^3-3, {x, -2-2I, 2+2I}, PlotRange->{0,10}]
```

3. С помощью команд Timing и AbsoluteTime установите, сколько времени уходит на нахождение корней уравнения x^7+x^3-3=0 со 100 значащими цифрами. Очистите память с помощью команды меню Evaluation/Quit Kernel/Local, а затем выполните свою программу дважды. Чем объяснить различное время счета?

```
1)Timing[N[Solve[x^7+x^3-3==0,x], 100];]
a = AbsoluteTime[]
N[Solve[x^7+x^3-3==0,x], 100];
b= AbsoluteTime[]
     b - a
     2)AbsoluteTime[]
```

4. Присвойте последовательным переменным x1, x2, ..., x7 значения корней уравнения x^7 + x^3 - 3 = 0 (можно использовать команду Do).

```
root=N[Solve[x^7+x^3-3==0,x]]
Do[
 Subscript[x,i]=root[[i]],{i,7}]
Subscript[x,7]
```

5. С помощью команд Prime и Table составьте список первых 100 простых чисел. С помощью команды Part найдите 15-ое простое число. С помощью команды Position найдите порядковый номер простого числа 127.

```
a=Table[Prime[i],{i,1,100}]
Part[a,15]
Part[Position[a,127],1,1]
```

6. Создайте список {3, 5, 7, 9, 11} с помощью команды Range. Создайте диагональную матрицу, на диагонали которой стоят перечисленные числа (команда DiagonalMatrix).

```
diagonal=Range[3,11,2]
dd=DiagonalMatrix[diagonal];
MatrixForm[dd]
```

7. Напишите команду prepend0, дописывающую к каждому списку впереди и сзади ноль (используйте Append и Prepend).

```
vector=List[a,b,s,d]
Prepend[Append[vector,0],0]
```

8. Чем отличаются возможности команд ListPlot и DiscretePlot?Изобразите множество точек (n, sin n/ 10), когда n меняется от 0 до 50.

$$\text{DiscretePlot}[\text{Sin}[n/10],\{n,0{,}50\}]$$

$$\text{ListPlot}[\text{Table}[\text{Sin}[n/10],\{n,0{,}50\}]]$$

9. Создайте список f[3], f[4], ..., f[11] с помощью команды Array. Преобразуйте этот список в сумму f[3]+f[4]+...+f[11] с помощью команды Apply.

```
list=Array[f,9,3]
Plus @@list
```

10. Напишите программу, превращающую целое число 255629 в список его цифр {2,5,5,6,2,9}. УКАЗАНИЕ . Используйте команды FractionalPart , IntegerPart , Reverse , Log10 , Table . Ср. с работой команд IntegerDigits и RealDigits.

```
n=255629
a=FractionalPart[n/10^k];
b=IntegerPart[a*10];
Table[b,{k,1,Log10[n]+1}]//Reverse
```

11. Составьте программу, превращающую список натуральных чисел в True, если в нем есть простые числа, и в False в противном случае (используйте команды PrimeQ, Apply, Or).

```
x={2,3,4,5};
y={4,6,8};
a=PrimeQ[x];
Apply[Or,a]
b=PrimeQ[y];
Apply[Or,b]
```

12. Напишите программу, вычисляющую угол между заданными векторами (используйте команду Norm)

```
n=2;
u=Table[Random[],{n}]
v=Table[Random[],{n}]
Dot[u,v]/(Norm[u]*Norm[v])
```

13. Найдите наибольшее отрицательное число в заданном списке (используйте команды Select , Negative , Max).

```
list={1, -15, -6, 0,-2};
Max[Select[list,Negative]]
```

14. Дана матрица и строка (столбец).Вставьте строку (столбец) на k-ое место (команды Insert и Transpose).

```
n=5;
m=Table[i-j,{i,n},{j,n}];
MatrixForm[m]
s=Table[Random[],{n}]
d=Insert[m,s,4];
MatrixForm[d]
res=Transpose[Insert[Transpose[m],s,4]];
MatrixForm[res]
```

15. Для n = 4 создайте матрицу размера nЧn, состоящую из (одних и тех же) чисел 3 (команда Table или ConstantArray). Замените (команды Part и Do) в ней наддиагональные элементы на случайные числа из отрезка[0, 10] (команда Random), имеющие две значащие цифры после запятой (команда NumberForm) Выведите окончательный результат на экран командой MatrixForm. А как поменять поддиагональные элементы? Сделайте то же самое для n = 5.

```
n=4;
d=ConstantArray[3,{n,n}]
MatrixForm[d]
Do[d[[i,j]]=NumberForm[Random[],{2,1}],{i,n-1},{j,i+1,n}];
MatrixForm[d]
```

16. Изобразите график функции y=x/(x^2-1) и его асимптоты. Указание. Можно использовать команды Show и ParametricPlot.

```
g1=Plot[x/(x^2-1),{x,-3,3},PlotStyle->{Blue}];
g2=ParametricPlot[{{1,y},{-1,y}},{y,-4,4},PlotStyle-
>{{Black,Dashed}}];
Show[g1,g2]
```

17. Сформируйте массив из n=5 точек вида (x_i,f(x_i)), где f(x)=Sin[x]. Изобразите их с помощью команды ListPlot. Сделайте точки большими (PointSize) и яркими. На том же рисунке изобразите график функции f и полученные точки (Show). Постройте по имеющимся точкам интерполяционный многочлен (InterpolatingPolynomial) и

изобразите его график на том же рисунке другим цветом и пунктиром. Как меняется рисунок при увеличении n? Какое максимальное значение n можно взять?

```
n=5;
x=Range[0,2 Pi,2 Pi/n]//N;
f[t_]=Sin[t];
list=Table[{x[[i]],f[x[[i]]]},{i,1,Length[x]}];
point=ListPlot[list,PlotStyle->{Black,PointSize[Large]}];
g1=Plot[f[t],{t,x[[1]]-1,Last[x]+1}];
pol[t_]=InterpolatingPolynomial[list,t];
gpol=Plot[pol[t],{t,x[[1]]-1,Last[x]+1},PlotStyle->{Red,Dashed}];
Show[point,g1,gpol,PlotRange->All]
```

18. Создайте массив из n=100 случайных точек, принадлежащих [0,1]Ч[0,1], каждая координата которых имеет равномерное распределение (команда RandomReal[]). Изобразите их с помощью команды ListPlot. Похоже ли распределение на равномерное? Увеличьте n до 10000.

```
n=100;
randomNumbers=RandomReal[{0,1},{n,2}];
ListPlot[randomNumbers]
```

19. Сделайте мультипликацию, на которой вращается кривая r=Sin[3ϕ].
$$\mathrm{Animate}[\mathrm{PolarPlot}[\mathrm{Sin}[3*\mathrm{phi}+a],\{\mathrm{phi},0,\mathrm{Pi}\},\mathrm{PlotRange}-\\ >1],\{a,0,\mathrm{Pi}\},\mathrm{AnimationRunning}->\mathrm{False}]$$

20. Найдите корни многочлена z^5+a*z-1 (используйте Solve) при a=1. Изобразите их на комплексной плоскости (ListPlot). Сделайте мультипликацию зависимости корней от a (команда Animate или Manipulate).

```
roots=NSolve[z^5+a*z-1==0,z];
r1=z/.roots ;
r2:={Re[r1],Im[r1]};
r3 :=Transpose[r2]
ListPlot[r3/. a->1]
Animate[ListPlot[Transpose[{Re[z/. NSolve[z^5+a*z-1==0,z]],Im[z/.
NSolve[z^5+a*z-1==0,z]]}],PlotRange->{{-2,2},{-2,2}}],{a,-
3,3},AnimationRunning->False]
```

21. Посчитайте первые n=5 производных функции y=Exp[-x^2/2].

```
n=5;
s=Table[D[Exp[-x²/2],{x,i}],{i,n}];
TableForm[s]
```

22. Разложите в сумму действительных элементарных дробей функцию y= (x^7+x^4-13)/(x^8+x^7+2x^6+2x^5+x^4+x^3). Используйте команду Apart.
23. Найдите многочлен наибольшей степени, являющийся делителем многочленов p=z^210-1 и q=z^90-1 (команда PolynomialGCD или PolynomialLCM).

24. Разложите многочлен z^7+z^6+64z+64 на *линейные* комплексные множители (команда Factor, опция Extension). Объясните результат работы команды Factor[x^50-1, Extension □ I](пытается разложить выражение x^50 - 1 на множители, используя расширение поля комплексных чисел.). Разложите многочлен z^7+z^6+64z+64 на линейные и квадратичные действительные множители.

```
Factor[z^7+z^6+64 z+64]
Factor[z^7+z^6+64 z+64,Extension->I]
```

25. Представьте рациональную функцию y=(z^6-z^2+1)/(z^3+1) в виде суммы многочлена и *правильной* рациональной функции (команды Numerator, Denominator, PolynomialQuotient, PolynomialRemainder).

$y=(z^6-z^2+1)/(z^3+1)$

```
n=Numerator[y];
d=Denominator[y];
m=PolynomialQuotient[n,d,z];
f=PolynomialRemainder[n,d,z];
m+f/d
```

26. Найдите сумму всех коэффициентов многочлена (4z^2-2z-1)^(13)(5z^2-7)^4

```
p1=(4z^2-2z-1)^(13)(5z^2-7)^4;
p2=Expand[p1];
c=CoefficientList[p2,z];
sum=Apply[Plus,c]
```

27. Для n=7 найдите n-ый многочлен Маклорена функции y=Cos[x]*Sin[9x]. Изобразите на одном рисунке график исходной функции и ее многочлена Маклорена. Как меняется рисунок при увеличении n? Попробуйте сделать мультипликацию.

```
y1[x_]=Cos[x]*Sin[9 x];
y2[x_,n_]:=Series[Cos[x]*Sin[9 x],{x,0,n}]//Normal
ListAnimate[Table[Plot[{y1[x],Evaluate[y2[x,n]]},{x,-1,1}],{n,7}]]
```

28. Посчитайте многочлен Маклорена степени n=5 функции, обратной к y= $\int_{-\infty}^{x} \mathrm{Exp}[-t^{\wedge}2/2]\, dt$ (команда InverseSeries).

$y=\int_{-\infty}^{x} e^{-t^2/2}\, dt$;

```
r=InverseSeries[Series[y,{x,0,5}]]//Normal
```

29. Напишите команду, раскладывающую многочлен по степеням x-a.

```
p=x^4-3x^2+x-2
Series[p,{x,a,5}]//Normal
```

30. Чем отличаются опции StepMonitor и EvaluationMonitor? На примере нахождения корня уравнения arctg x+(cos10x)/10=0 с помощью команды FindRoot методом секущих (опция Secant) продемонстрируйте различие между этими двумя опциями.

основное отличие между опциями StepMonitor и EvaluationMonitor заключается в том, что StepMonitor отслеживает каждый шаг вычисления, в то время как EvaluationMonitor отслеживает только процесс вычисления всего выражения или функции.
значение каждого вхождения в цикл 2.отслеживает значение ф-ции на каждом шаге

```
f[x]=ArcTan[x]+Cos[10 x]/10;
result=FindRoot[f[x],{x,1},Method-
>"Secant",StepMonitor:>Print[x]]

f[x]=ArcTan[x]+Cos[10 x]/10;
result1=FindRoot[f[x],{x,1},Method-
>"Secant",EvaluationMonitor:>Print[x]]
```

31. Для заданного натурального n изобразите график функции y= $\sum_{k=1}^{n}(\mathrm{Sin}[x])^\wedge k$ (команды Sum и Plot). Как меняется график с изменением n? Сделайте мультипликацию (ListAnimate), показывающую изменение графика с ростом n.

```
n=5;
ListAnimate[Table[Plot[Sum[Sin[x]^i,{i,k}],{x,0,4 Pi},PlotRange-
>n],{k,n}]]
```

32. Сформируйте список a из n=100 случайных целых чисел, лежащих в отрезке [-1000,1000] (команда RandomInteger). Составьте новый список, в который входят в точности положительные члены списка a (команды Delete, Append и Do или DeleteCases).

```
n=100;
a=Table[RandomInteger[{-1000,1000}],{n}]
b={};
Do[If[a[[i]]>0,AppendTo[b,a[[i]]]],{i,n}];
Print[b]
```

33. Для заданного натурального n составьте матрицу, на главной диагонали которой стоят числа от 1 до n (Range, DiagonalMatrix), а на соседних диагоналях --- числа от n-1 до 1. Найдите и изобразите (Eigenvalues, ConplexListPlot) собственные значения. Как меняется спектр при увеличении n?

спектр расширяется число собственных значений увеличивается

```
n=5;
a=DiagonalMatrix[Range[n]];
Do[a[[i+1,i]]=a[[i,i+1]]=n-i,{i,n-1}]
MatrixForm[a]
ei=Eigenvalues[a];
w=Transpose[{Re[ei],Im[ei]}];
  ListPlot[w]
```

34. Напишите программу (оформленную как Module), находящую наибольшее в данном списке действительных чисел. Добавьте проверку того, что элементы списка действительно являются числами (используя четвертый аргумент команды If). Сравните с работой команды Max.

```
x1=Table[Random[],{10}]
Print["вывод работы стандартной команды Max:",Max[x1]]
AppendTo[x1,100+I]
max[x0_]:=Module[{x=x0,max},max=x0[[1]];

Do[If[MatchQ[x[[i]],_Real],If[x[[i]]>max,max=x[[i]]],Print[x[[i]]
," - не является действительным
числом"]],{i,2,Length[x]}];Print["вывод работы команды
max:",max]]
max[x1]
```

35. Запрограммируйте вычисление HeavisidePi(x) с помощью команды Piecewise. Проверьте результат путем построения графика.

```
p1=HeavisidePi[x];
HeavisidePi[1/3]
Plot[p1,{x,-2,2}]
h[x_]=Piecewise[{{1,Abs[x]<1/2},{0,Abs[x]>1/2}}]
h[1/3]
Plot[h[x],{x,-2,2},PlotStyle->{Green}]
```

36. Найдите первое из чисел последовательности x_n=Sin[n], для которого $|x_n|<10^{-4}$ (используйте команду While). Выведите на печать соответствующее n.

```
n=1;
While[Abs[Sin[n]]>=10^(-4),n++];
Print["n=",n]
```

37. Вычислите e^{-20} путем подстановки x=-20 в ряд Маклорена для экспоненты. Сравните результат с $1/e^{20}$. Чем объяснить расхождение?

```
x=-20;
N[Sum[(x)^i/i!,{i,0,∞}]]
N[1/Exp[20]]
```

38. Интегральным оператором Фредгольма называют оператор (Ku)(x)= $\int_a^b k(x,y)\,u(y)\,dy.$ Функцию k называют его ядром. Ядро произведения $K=K_1K_2$ двух операторов Фредгольма находят по формуле k[x,y]= $\int_a^b k_1(x,t)\,k_2(t,y)\,dt.$ Напишите программу (оформленную как Module) вычисляющую ядро произведения двух операторов Фредгольма. Как проверить правильность работы программы?

```
F[k1_,k2_,a0_,b0_]:=Module[{a=a0,b=b0},k3[x_,y_]=Integrate[k1[x,t
]*k2[t,y],{t,a,b}];Print[k3[x,y]]];
```

```
k[x_,y_]=x*y;
a=0;
b=1;
F[k,k,a,b];
```

39. Интегральным оператором Вольтерра называют оператор (Ku)(x)= $\int_a^x k(x,\,y)\,u(y)\,dy.$ Функцию k называют его ядром. Ядро произведения $K=K_1K_2$ двух операторов Вольтерра находят по формуле k[x,y]= $\int_y^x k_1(x,\,t)\,k_2(t,\,y)\,dt.$ Напишите программу (оформленную как Module) вычисляющую ядро произведения двух операторов Вольтерра.

```
V[k1_,k2_,a0_]:=Module[{a=a0},k3[x_,y_]=Integrate[k1[x,t]*k2[t,y]
,{t,y,x}];Print[k3[x,y]]];
k[x_,y_]=x*y;
V[k,k,1]
```

40. Выведите на экран первые 4 члена последовательности $z_1=1+x$, $z_{n+1}=1+\sqrt{z_n}$. Используйте команду For.

$$\mathrm{For}[z = x + 1; \mathrm{Print}[z]; n = 1, n < 4, z = 1 + \mathrm{Sqrt}[z]; n + +; \mathrm{Print}[z]]$$

41. Последовательность z_n удовлетворяет разностному уравнению $z_{n+1}=1+z_n-z_{n-1}$ и начальным условиям $z_0=2$ и $z_1=1$. Постройте график первых 20 членов, вычислите их произведение (Product).

```
np=20;
z=Table[0,{np}];
z[[1]]=2;z[[2]]=1;
For[{n=3},n<np,n++,z[[n]]=1+z[[n-1]]-z[[n-2]]]
ListPlot[z]
Product[z[[k]],{k,np}]
```

42. Схема Горнера вычисления многочлена $b_0+b_1x+...+b_mx^m$ может интерпретироваться как значение решения s_0 неоднородного разностного уравнения $s_{i-1}=x*s_i+b_{i-1}$, i=m, m-1, ..., 1, с начальным условием $s_m=b_m$. Напишите программу (оформленную как Module), вычисляющую значение многочлена в заданной точке x по списку его коэффициентов.

```
polyValue[x_,b_List]:=Module[{n=Length[b],s},s=b[[n]];
  For[i=n-1,i>=1,i--,s=x*s+b[[i]];];s]
coefficients={2,-3,1,0,5};
xValue=2;
result=polyValue[xValue,coefficients];
Print[result]
```

43. Используя операторы Set и SetDelayed (попробуйте оба варианта и сравните результаты), напишите команду, вычисляющую плотность нормального распределения

ф(x)=1/(σ $\sqrt{2\pi}$) $e^{-(x-a)^2/(2\sigma^2)}$ с опциальными параметрами a=0 и σ=1. Используйте команду Optional.

```
phi1[x_,sigma_:1,a_:0]:=Exp[-(x-a)^2/(2 sigma^2)]/(sigma*Sqrt[2 Pi])
phi1[t, z]
```

```
phi2[x_,sigma_:1,a_:0]=Exp[-(x-a)^2/(2 sigma^2)]/(sigma*Sqrt[2 Pi]);
phi2[t]
```

44. Задайте функцию y= e^{-x^2} анонимным способом. Вычислите ее значение при x=3.

```
Exp[-#^2]&;
%[3]
```

45. Дан список {1,2,3}. Поместите каждый элемент в фигурные скобки: {{1},{2},{3}}. Используйте команду Map.

```
a={1,2,3}
Map[List,a]
```

46. Дан список чисел. Замените элементы списка с четными порядковыми номерами на их квадраты. Используйте команду MapAt.

```
a={1,2,3,4,5,-1,4,5};
f[x_]:=x^2
n=Range[2,Length[a],2]
m=Map[List,n]
MapAt[f,a,m]
```

47. Дан список переменных {a,b,c} и список чисел {1,2,3}. Создайте список команд {a:>1,b:>2,c:>3}. Используйте Thread.

$$\text{Thread}[\{a, b, c\} :> \{1,2,3\}]$$

48. Дано семейство функций f_1,f_2,f_3. Составьте список, в котором каждая функция применена к одному и тому же аргументу t (используйте Map и анонимную функцию).

Дано семейство функций f_1,f_2,f_3 (зависящих от трех переменных) и список аргументов x_1,x_2,x_3. Составьте список, в котором каждая функция применена к одному и тому же набору аргументов x_1,x_2,x_3 (используйте Map и Sequence).

Дано семейство функций f_1,f_2,f_3(зависящих от одной переменной) и список аргументов x_1,x_2,x_3. Составьте список, в котором каждая функция применена к своему аргументу (используйте MapThread).

```
functions={Subscript[f,1],Subscript[f,2],Subscript[f,3]};
argument=t;
result=Map[#[argument]&,functions]
```

```
functions={Subscript[f,1],Subscript[f,2],Subscript[f,3]};
arguments={Subscript[x,1],Subscript[x,2],Subscript[x,3]};
result=Map[#[Sequence@@arguments]&,functions]
```

```
functions={Subscript[f,1],Subscript[f,2],Subscript[f,3]};
arguments={Subscript[x,1],Subscript[x,2],Subscript[x,3]};
result=MapThread[#1[#2]&,{functions,arguments}]
```

49. Дан список векторов, образующих базис. Вычислите матрицу Грама --- матрицу всевозможных попарных скалярных произведений. Используйте Outer с четырьмя аргументами.

```
vectors={{1,2},{3,4+I}};
gramMatrix=Outer[Dot,vectors,Conjugate[vectors],1];
MatrixForm[gramMatrix]
```

50. Напишите команду, вычисляющую данную степень n данной матрицы a. Используйте Nest. Придумайте тестовые примеры.

```
power[a_,n_]:=Nest[Dot[a,#]&,IdentityMatrix[Length[a]],n]
a={{1,2},{3,4}};
p=power[a,2];
MatrixForm[p]
```

51. Дан список коэффициентов aa={a_0,...,a_n}. Напишите функцию, вычисляющую многочлен с этими коэффициентами. Используйте Fold. Что произойдет, если в эту функцию вместо переменной x подставить матрицу?

```
poly[a_,x_]:=Fold[#1*x+#2&,0,Reverse[a]]//Expand
coefficients={Subscript[a,0],Subscript[a,1],Subscript[a,2]};
x=d;
result=poly[coefficients,x]
```

```
poly[a_,x_]:=Fold[#1*x+#2&,0,Reverse[a]]//Expand
coefficients={Subscript[a,0],Subscript[a,1],Subscript[a,2]};
n=3;
m=Table[i-j,{i,n},{j,n}];
x=MatrixForm[m]
result=poly[coefficients,x]
```

52. Напишите функцию, превращающую список цифр {2, 5, 5, 6, 2, 9} в целое число 255629. Используйте Fold.

```
toInteger[list_] := Fold[#1*10 + #2 &, 0, list]
toInteger[{2,5,5,6,2,9}]
```

53. Напишите команду, преобразующую список {{a,3},{b,2},{c,1},{d,5}} в {a,a,a,b,b,c,d,d,d,d,d}.

```
d[l_List]:=Flatten[Table[ConstantArray[l[[i,1]],l[[i,2]]],{i,Leng
th[l]}]]
l={{a,3},{b,2},{c,1},{d,5}};
```

```
d[l]

2 вариант: d[l_List]:=Module[{n=Length[l]},s={};

Do[AppendTo[s,ConstantArray[l[[i,1]],l[[i,2]]]],{i,n}];s//Flatten
]
l={{a,3},{b,2},{c,1},{d,5}};
d[l]
```

54. Напишите шаблоны, означающие Log от любого аргумента; переменная x в целочисленной степени; сумма двух выражений; непустой список; произвольный список, включая пустой. Проверьте правильность с помощью команды MatchQ.

```
MatchQ[Log[5],Log[_]]
MatchQ[x^8,x^n_Integer]
MatchQ[7+3a,x_+y_]
MatchQ[{x,a},List[__]]
MatchQ[{},List[___]]
```

55. В тригонометрическом выражении, содержащем косинусы и синусы (например, Cos[x/3]^2+Sin[y]^2+Log[x]), замените все аргументы на половинные. Используйте команду ReplaceAll.

$\mathrm{Cos}[x/3]\hat{}2 + \mathrm{Sin}[y]\hat{}2 + \mathrm{Log}[x]/.\{\mathrm{Sin}[x\_]-> \mathrm{Sin}[x/2], \mathrm{Cos}[x\_]-> \mathrm{Cos}[x/2], \mathrm{Log}[x]-> \mathrm{Log}[x/2]\}$

56. Дан список z={1,2,0,3,4,s,g,j,k,4,2,1+I*7}. Посчитайте число ненулей в списке (Count, PatternTest или Except), количество комплексных чисел в списке; отберите элементы, являющиеся четными числами (Cases, PatternTest, EvenQ); удалите четные числа из списка (DeleteCases, PatternTest, EvenQ); выпишите позиции чисел, больших 0 (Position, PatternTest).

```
z={1,2,0,3,4,s,g,j,k,4,2,1+I*7};
Count[z,Except[0]]
Count[z,_Complex]
Cases[z,_?EvenQ]
DeleteCases[z,_?EvenQ]
Position[z,_?(Abs[#]>0&)]
```

57. Определите функцию, равную Cos при положительных значениях аргумента, и Cosh --- при отрицательных. Используйте команду: 1) Condition, 2) Piecewise. Постройте график. Распространите на эту функцию действие операции дифференцирования (TagSet).

```
f[x_]:=Cos[x]/;x>0
f[x_]:=Cosh[x]/;x<=0
Plot[f[x],{x,-4,4}]
g[x_]=Piecewise[{{Cos[x],x>0},{Cosh[x],x<=0}}]
Plot[g[x],{x,-4,4}]
f/:D[f[x_],x_]:=Piecewise[{{-Sin[x],x>0},{Sinh[x],x<=0}}]
D[f[x],x]
```

58. Определите функцию, в которую можно подставлять любое ненулевое число аргументов и которая вычисляет среднее арифметическое с весами 1/k.

```
mean[z0__]:=Module[{s,z},z=List[z0];
  s=Sum[z[[i]]/i,{i,Length[z]}]/Sum[1/i,{i,Length[z]}]]
mean[a,b,c]
```

59. Напишите команду change с опцией name -> "Plus" или name -> "Product", превращающую выражение f[x] в выражение Plus[x] или Product[x]. Если опция name не задана, ответом должно быть x. Объясните действие команды change на g[p,q] и на Exp[q].

```
change[f_[x__],OptionsPattern[name-
>"Automatic"]]:=Switch[OptionValue[name],"Automatic",Print[x],"Pl
us",Plus[x]];
change[f[1,2]]
change[f[1,2],name->"Plus"]
change[f[g[p,q]],name->"Plus"]
Plus[g[p,q]]
change[f[Exp[q]],name->"Plus"]
Plus[Exp[q]]
```

60. Для данного n сформируйте список z, состоящий из n *различных* чисел. Дана функция f. *Разделенные разности* функции f образуют таблицу. В первой строке стоят значения функции $f_i^{[0]}$=f(z_i). Вторая строка состоит из чисел $f_i^{[1]}$=($f_{i+1}^{[0]} - f_i^{[0]}$)/($z_{i+1}$-$z_i$). Следующая строка строится по второй по тому же правилу: $f_i^{[2]}$=($f_{i+1}^{[1]} - f_i^{[1]}$)/($z_{i+1}$-$z_i$). И т.д. Постройте таблицу из разделенных разностей. Оформите результат в виде Module. Как проверить правильность работы?

```
divDif[f0_,w_]:=Module[{z=w,f=f0,dD},
  dD[f_,z_,i_,j_]:=0/;i<1;
  dD[f_,z_,i_,j_]:=0/;i>j;
  dD[f_,z_,i_,j_]:=f[z[[i]]]/;i==j;
  dD[f_,z_,i_,j_]:=(dD[f,z,i+1,j]-dD[f,z,i,j-1])/(z[[j]]-
z[[i]])/;i<j;
  n=Length[z];
  d=Table[dD[f,z,j,j+i],{i,0,n},{j,1,n-i}];
  TableForm[d]]
divDif[F,z=Table[Subscript[Z,i],{i,3}]]
```

61. Двумерное нормальное распределение задается плотностью вероятности `f[x_,y_]:=1/(2*Pi*sigma1*sigma2*Sqrt[1-cor^2])*Exp[-1/(2*(1-cor^2))*((x-a1)^2/sigma1^2+(y-a2)^2/sigma2^2-2*cor*((x-a1)*(y-a2))/(sigma1*sigma2))]`. А двумерное полиномиальное распределение имеет распределение вероятности `pr[k_,l_,m_]=Multinomial[k,l,m]` p^k q^l r^m, где r:=1-p-q, m:=n-k-l. Постройте графики этих функций (команды Plot3D и ListPointPlot3D). Для p=1/2, q=1/3 и n=50 подберите параметры нормального распределения так, чтобы f приближала pr (двумерный аналог локальной теоремы Муавра--Лапласа). Указание. Рассмотрите зависимость pr только от k и l. Вначале покажите, что математическим

ожиданием полиномиального распределения является (np,nq). Затем вычислите выборочные дисперсии и коэффициент корреляции и возьмите их в качестве sigma1, sigma2 и cor.

```
f[x_,y_]:=1/(2*Pi*sigma1*sigma2*Sqrt[1-cor^2])*Exp[-1/(2*(1-
cor^2))*((x-a1)^2/sigma1^2+(y-a2)^2/sigma2^2-2*cor*((x-a1)*(y-
a2))/(sigma1*sigma2))]
r:=1-p-q;
m:=n-k-l;
pr[k_,l_,m_]:=Multinomial[k,l,m] *p^k* q^l *r^m;
meanK=Sum[k*pr[k,l,m],{k,0,n},{l,0,n-k}]//Simplify
meanL=Simplify[Sum[l*pr[k,l,m],{k,0,n},{l,0,n-k}],Assumptions-
>Element[n,Integers]]
n=50;
p=1/2;
q=1/3;
a1=n*p;
a2=n*q;
varKK=Sum[(k-n*p)^2*pr[k,l,m]//N,{k,0,n},{l,0,n-k}];
varLL=Sum[(l-n*q)^2*pr[k,l,m]//N,{k,0,n},{l,0,n-k}];
varKL=Sum[(k-n*p)*(l-n*q)*pr[k,l,m]//N,{k,0,n},{l,0,n-k}];
sigma1=Sqrt[varKK];
sigma2=Sqrt[varLL];
cor=varKL/(sigma1*sigma2);
tab=Table[{k,l,pr[k,l,m]//N},{k,0,n},{l,0,n-k}];
ff1=ListPointPlot3D[tab,PlotRange->All];
ff2=Plot3D[f[x,y],{x,0,n},{y,0,n},PlotRange->All,PlotPoints->50];
Show[ff1,ff2]
```

62. Напишите команду, вычисляющую градиент заданной функции n переменных (используйте Apply или Outer}). Ср. с работой встроенной команды Grad. Затем напишите команду, вычисляющую по заданной функции n переменных и заданному n-мерному вектору производную этой функции в направлении этого вектора.

```
f[x_,y_,z_]=x+2y^2+z^4;
grad[f_,n_]:=Map[D[f[Apply[Sequence,Table[Subscript[x,i],{i,n}]]]
,#]&,Table[Subscript[x,i],{i,n}]]
grad[f,3]
Grad[f[x,y,z],{x,y,z}]
vec={1,2,3};
vec.Map[D[f[Sequence@@Table[Subscript[x,i],{i,Length[vec]}]],#]&,
Table[Subscript[x,i],{i,Length[vec]}]]
```

63. Создайте графики суперпозиций всех тригонометрических функций и обратных к ним в том и другом порядке. Используйте Thread и Composition. Почему графики не совпадают с графиком функции y=x?
В первом случае графики совпадают с графиком функции y=x, потому что композиция обратной и прямой функций даёт исходную переменную. То есть, если

мы берем функцию f(x), затем применяем к ней обратную функцию g(x), а затем снова применяем функцию f(x), мы получим исходную переменную x.

Во втором случае графики не совпадают с графиком функции y=x, потому что композиция прямой и обратной функций не обязательно даёт исходную переменную.потому что обратная функция не всегда существует или не всегда является однозначной.

```
direct={Cos,Sin,Tan,Cot}
inverse={ArcCos,ArcSin,ArcTan,ArcCot}
Map[Plot[#[x],{x,-10 Pi,10
Pi}]&,Thread[Composition[direct,inverse]]]
Map[Plot[#[x],{x,-10 Pi,10 Pi}]&,Thread[Composition[inverse,direct]]]
```

64. Переопределите Sin[x] при |x|<1 как тождественный ноль. Составьте таблицу значений полученной функции на отрезке [-3,3] с шагом 0.5. Нарисуйте график полученной функции. Чем объяснить расхождение?
    Расхождение объясняется тем, что при переопределении Sin[x] при |x|<1 как тождественный ноль, мы фактически заменяем значения синуса на нули в этом диапазоне. Это приводит к изменению формы графика и его поведения в окрестности точки x=0.

```
Unprotect[Sin]
Sin[x_]:=0/;Abs[x]<1;
table=Table[{x,Sin[x]},{x,-3,3,0.5}];
ListPlot[table,Joined->True,PlotMarkers->Automatic]
```

65. Из выражения 1+2+3 получите список слагаемых {1,2,3}, используя команды Apply, Hold, ReleaseHold.

$$\text{ReleaseHold}[\text{Apply}[\text{List}, \text{Hold}[1 + 2 + 3], \{1\}]]$$

66. Из выражения 4 + 5 + 6 получите (вычисленное) произведение 4Ч5Ч6, используя команды Apply, Unevaluated, Evaluate.

```
Unevaluated[4+5+6]
r=Apply[Times,%,{1}]
Apply[Evaluate,r]
```

67. Напишите команды, вычисляющие матрицу Вронского и вронскиан списка функций $\{f_1,f_2,f_3\}$. Используйте Outer, Det.

```
Clear[f,x]
t1=Table[Subscript[f,i],{i,3}]
m=Map[#[x]&,t1]
wronskianMatrix=Outer[D[#2,{x,#1}]&,Range[0,Length[t1]-1],m]
%//MatrixForm
wronskian=Det[wronskianMatrix]
```

68. Напишите команду вычисляющую вронскиан семейства функций ff={$f_{1,\dots,}$ f_n}. Используйте NestList.

```
Clear[f,x];
n=3;
functions=Table[Subscript[f,i],{i,n}];
wronskianMatrix=NestList[Map[(D[#,x]&),#1]&,Map[(#[x])&,functions
],n-1];
MatrixForm[wronskianMatrix]
jacobian=Det[wronskianMatrix]
```

69. Напишите команды, вычисляющие матрицу Якоби и якобиан списка функций {f_1,f_2,f_3}. Используйте Outer, Det.

```
Clear[f,x]
functions=Table[Subscript[f,i],{i,3}]
arg=Table[Subscript[x,i],{i,3}]
fff=Map[#[arg/. List->Sequence]&,functions]
jacobianMatrix=Outer[D,fff,arg];
MatrixForm[jacobianMatrix]
jacobian=Det[jacobianMatrix]
```

70. Напишите команду, вычисляющую дивергенцию трехмерного векторного поля {f_1,f_2,f_3}. Используйте Inner. Проверьте правильность, сравнивая результат со встроенной командой Div.

```
Clear[f,x]
t=Table[Subscript[f,i],{i,3}];
a=Table[Subscript[x,i],{i,3}];
ff=Map[#[a/. List->Sequence]&,t];
div[vec_,vars_]:=Inner[D,vec,vars,Plus]
div[ff,a]
Div[ff,a]
```

Пакеты:

1. Напишите пакет (расширение *.m), содержащий три команды ComplexToCartesian, превращающую список комплексных чисел в список их действительных и мнимых частей,
2) ListComplexPlot, изображающую список комплексных чисел и
3) EigenPlot, изображающую спектр матрицы.

Сделайте проверку.

Содержимое пакета:

```
BeginPackage["ComplexNumbers`"];
ComplexToCartesian::usage = "ComplexToCartesian[z]
превращает список z комплексных чисел в список
их мнимых и действительных частей";
ListComplexPlot::usage = "ListComplexPlot[z]
изображает список z комплексных чисел";
```

```
EigenPlot::usage = "EigenPlot[a] изображает
спектр матрицы a";
Begin["`Private`"];
ComplexToCartesian = Function[z, {Re[z], Im[z]},
   Listable];
ListComplexPlot[z_] :=
      ListPlot[ComplexToCartesian[z]];
EigenPlot[z_] :=
      ListComplexPlot[Eigenvalues[z]];
End[];
EndPackage[]
```

Код в вольфрам:

```
l={2-4I,1-3I,-2I};
ComplexToCartesian[l]
 ListComplexPlot[l]
 EigenPlot[{{1,2},{3,I}}]
```

2. Напишите пакет, содержащий команды,

вычисляющие нормы матрицы $A_{1\to1}=\max_j \sum_i |a_{ij}|$,

$A_{2\to2}=\sqrt{\lambda_{\max}(A^* A)}$

(квадратный корень из наибольшего собственного значения матрицы A^* A) и

$A_{\infty\to\infty}=\max_i \sum_j |a_{ij}|$. Сделайте проверку.

Содержимое пакета:

```
BeginPackage["MatrixNorm`"];
MatrixNorm1::usage = "MatrixNorm1[a] вычисляет норму
матрицы a при условии,
что в пространстве задана норма с индексом 1";
MatrixNorm2::usage = "MatrixNorm2[a] вычисляет норму
матрицы a при условии,
что в пространстве задана норма с индексом 2";
MatrixNormInf::usage = "MatrixNormInf[a] вычисляет норму
матрицы a при условии,
что в пространстве задана норма с индексом Infinity";
Begin["`Private`"];
MatrixNorm1[a_] :=
      Max[Total[Abs[a], {2}]];

MatrixNorm2[a_] :=
      Sqrt[Max[Eigenvalues[ConjugateTranspose[a] . a]]];

MatrixNormInf[a_] :=
      Max[Total[Abs[a], {1}]];
End[];
```

```
EndPackage[]
```

Код в вольфрам:

```
n=4;
matrix=Table[Random[],n,n]
MatrixNorm1[matrix]
MatrixNorm2[matrix]
MatrixNormInf[matrix]
```

3. Пусть A --- квадратная матрица. Если ||A||<1 (включите проверку этого условия, подгружая пакет из задания 11), то уравнение x+Ax=f можно решать методом последовательных приближений.

Напишите команду (оформленную как Module), решающую уравнение x+Ax=f методом последовательных приближений для данных A и f.

Код в вольфрам:

```
lS[a0_,f0_,max0_]:=Module[{a=a0 , f=f0, max=max0,x},
  x=f;

If[Max[MatrixNorm1[a],MatrixNorm2[a],MatrixNormInf[a]]>=1,Print["
Норма матрицы > 1"]];

If[Max[MatrixNorm1[a],MatrixNorm2[a],MatrixNormInf[a]]<1,Do[x=f-
a.x,max]];
  x
  ]


n=3;
a=RandomReal[{0.1,0.3},{n,n}] ;
f=RandomReal[{0.1,0.3},n];
x=lS[a,f,10]
     x+a.x-f
```