1. Sequences

```
> #Onepamop $
> restart :
> x$x = 1 ..10;
```

1, 2, 3, 4, 5, 6, 7, 8, 9, 10 **(1.1)**

```
> x^2$x = 1 ..10;
```

1, 4, 9, 16, 25, 36, 49, 64, 81, 100 **(1.2)**

```
> 2·x + 1$x = 1 ..10;
```

3, 5, 7, 9, 11, 13, 15, 17, 19, 21 **(1.3)**

```
> 2· x$x = 1 ..10;
```

2, 4, 6, 8, 10, 12, 14, 16, 18, 20 **(1.4)**

```
> x$ x = "A" .."Z";
```

"A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z" **(1.5)**

```
> x$x = "K" .."N";
```

"K", "L", "M", "N" **(1.6)**

2. Lists

```
> #Задание списков при помощи $
> restart :
> L := [ 1/n $n = 1 ..10 ];
```

$$L := \left[1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}, \frac{1}{10}\right] \quad \textbf{(2.1)}$$

```
> whattype(L);
```

$$list \quad \textbf{(2.2)}$$

```
> type(L, list);
```

$$true \quad \textbf{(2.3)}$$

3. Command seq

```
#Команда seq
restart :
q := seq(i, i = 1 ..5);
```

$$q := 1, 2, 3, 4, 5 \quad \textbf{(3.1)}$$

```
X := [seq(i, i = 0 ..6)];
```

$$X := [0, 1, 2, 3, 4, 5, 6] \quad \textbf{(3.2)}$$

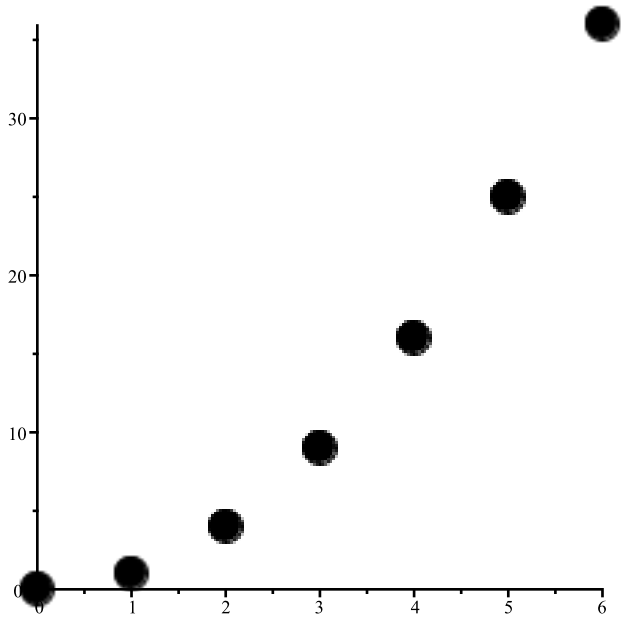
```
Y := [seq(i^2, i = X)];
```

$$Y := [0, 1, 4, 9, 16, 25, 36] \quad \textbf{(3.3)}$$

```
P := [seq([X[i], Y[i]], i = 1 ..nops(X))];
```

$$P := [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36]] \quad \textbf{(3.4)}$$

```
with(plots) : pointplot(P, symbol = solidcircle, symbolsize = 36);
```




```
seq(i, i = "Hello");
```

"H", "e", "l", "l", "o" **(3.5)**

```
seq(i, i = "a" .."f");
```

"a", "b", "c", "d", "e", "f" **(3.6)**

```
L := [seq(i, i = 1 ..10, 2)];
```

$$L := [1, 3, 5, 7, 9] \quad \textbf{(3.7)}$$

```
L := [seq(i, i = 10 ..1, -2)]
```

$$L := [10, 8, 6, 4, 2] \quad \textbf{(3.8)}$$

4. Sets

> *#Задание множеств при помощи $*

> $S := \left\{ n\$n = \frac{1}{3} .. \frac{10}{3} \right\};$

$$S := \left\{ \frac{1}{3}, \frac{4}{3}, \frac{7}{3}, \frac{10}{3} \right\} \quad (4.1)$$

> $whattype(S);$

$$set \quad (4.2)$$

> $type(S, set);$

$$true \quad (4.3)$$

5. Commands for lists

```
> #Команды работы со стпсками
> restart :
> L := [n$n = 1 ..10 ];
```

$$L := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] \quad \textbf{(5.1)}$$

```
> nops(L);
```

$$10 \quad \textbf{(5.2)}$$

```
> op(2 ..5, L);
```

$$2, 3, 4, 5 \quad \textbf{(5.3)}$$

```
> L := [op(L), 256000 ];
```

$$L := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 256000] \quad \textbf{(5.4)}$$

```
>
```

```
> #Удаление элементов списка в соответствии с заданным правилом
> L := remove(x→x > 5, L);
```

$$L := [1, 2, 3, 4, 5] \quad \textbf{(5.5)}$$

```
> L := remove(x→x = 1, L);
```

$$L := [2, 3, 4, 5] \quad \textbf{(5.6)}$$

```
> L := remove(x→irem(x, 2) = 0, L); # остаток от деления нацело
```

$$L := [3, 5] \quad \textbf{(5.7)}$$

```
> L := remove(x→iquo(x, 5) = 1, L); # деление нацело
```

$$L := [3] \quad \textbf{(5.8)}$$

> ***#Отбор элементов списка в соответствии с заданным правилом***

> $L := [n\$n = 1..10];$

$$L := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] \quad \textbf{(5.9)}$$

> $L := select(x \to irem(x, 2) = 0, L);$ *# остаток от деления нацело*

$$L := [2, 4, 6, 8, 10] \quad \textbf{(5.10)}$$

> $L := select(x \to x < 8, L);$

$$L := [2, 4, 6] \quad \textbf{(5.11)}$$

```
> #Отбор элементов списка по типу
> L := ["1", 2, 3, "4", 5];
```

$$L := ["1", 2, 3, "4", 5] \quad \textbf{(5.12)}$$

```
> L := select(type, L, string);
```

$$L := ["1", "4"] \quad \textbf{(5.13)}$$

```
> L := ["1", 2, 3, "4", 5];
```

$$L := ["1", 2, 3, "4", 5] \quad \textbf{(5.14)}$$

```
> L := select(type, L, numeric);
```

$$L := [2, 3, 5] \quad \textbf{(5.15)}$$

```
>
```

```
> #Объединение списков
> L1 := [1, 2, 3];
```

$$L1 := [1, 2, 3] \tag{5.16}$$

```
> L2 := [4, 5];
```

$$L2 := [4, 5] \tag{5.17}$$

```
> LL := [op(L1), op(L2)]
```

$$LL := [1, 2, 3, 4, 5] \tag{5.18}$$

```
>
```

> # ***zip. Действие со списками по заданному правилу***

> $L1;$

$$[1, 2, 3] \quad \textbf{(5.19)}$$

> $L2;$

$$[4, 5] \quad \textbf{(5.20)}$$

> $L3 := zip((x, y) \to (x, y), L1, L2);$

$$L3 := [1, 4, 2, 5] \quad \textbf{(5.21)}$$

> $zip((x, y) \to [x, y], L1, L2);$

$$[[1, 4], [2, 5]] \quad \textbf{(5.22)}$$

> $zip\left((x, y) \to \left(x^2, \frac{y}{10}\right), L1, L2\right);$

$$\left[1, \frac{2}{5}, 4, \frac{1}{2}\right] \quad \textbf{(5.23)}$$

> $zip((x, y) \to (x \cdot y), L1, L2);$

$$[4, 10] \quad \textbf{(5.24)}$$

```
> # sort. Сортировка (по заданному правилу)
> L3;
```

$$[1, 4, 2, 5] \quad \textbf{(5.25)}$$

```
> sort(L3);
```

$$[1, 2, 4, 5] \quad \textbf{(5.26)}$$

```
> sort(L3, (x, y) → is(x > y)) :
>
```

6. Commands for sets

```
> # Действия с множествами
> restart :
> s := {1, 1, 2, 1, 3};
```

$$s := \{1, 2, 3\} \quad (6.1)$$

```
> s := {evalf(-exp(1)), 1024, "Some String", var, [[1, 2], [3, 4]]};
```

$$s := \{1024, -2.718281828, \text{"Some String"}, var, [[1, 2], [3, 4]]\} \quad (6.2)$$

```
> whattype(s);
```

$$set \quad (6.3)$$

```
> L := [op(s)];
```

$$L := [1024, -2.718281828, \text{"Some String"}, var, [[1, 2], [3, 4]]] \quad (6.4)$$

```
> whattype(L);
```

$$list \quad (6.5)$$

```
> s1 := {x$x = 1..3};
```

$$s1 := \{1, 2, 3\} \quad (6.6)$$

```
> s2 := {x$x = 4..5};
```

$$s2 := \{4, 5\} \quad (6.7)$$

```
> s3 := {op(s1), op(s2)};
```

$$s3 := \{1, 2, 3, 4, 5\} \quad (6.8)$$

```
> s1 := {1, 2, 3, 4, 5};
```

$$s1 := \{1, 2, 3, 4, 5\} \quad (6.9)$$

```
> s2 := {1, 2, 3};
```

$$s2 := \{1, 2, 3\} \quad (6.10)$$

```
> s3 := union(s1, s2);
```

$$s3 := \{1, 2, 3, 4, 5\} \quad (6.11)$$

```
> s3 := intersect(s1, s2);
```

$$s3 := \{1, 2, 3\} \quad (6.12)$$

```
> s3 := minus(s1, s2);
```

$$s3 := \{4, 5\} \quad (6.13)$$