

Задания практикума, тема массивы и числа

Общие замечания

Требуется написать программу с полным контролем ошибок. По завершению работы терминал должен быть приведён в исходное состояние. Для работы с параметрами командной строки нужно использовать функции: *ParamCount* и *Paramstr*. Об этом можно прочитать здесь: <https://www.freepascal.org/docs-html/rtl/system/paramcount.html>.

Ко всем программам должны быть разработаны тесты демонстрирующие реакцию на корректный ввод, а так же некорректный.

Пробельные символы (не путать с символом пробел) могут включать в себя пробелы, табуляции, перевод строки и возврат каретки.

1 калькулятор длинной арифметики

В аргументах командной строке программе передаётся количество чисел (не более 1024) и выходная система исчисления (не более 70-ричной). Далее в стандартном потоке ввода указываются числа в 70-ричной системе исчисления. Длина каждого числа не более 1000 цифр.

Для кодирования цифры в этой системе исчисления применяется алфавит:

[0..9] [a..z] [A..Z] [!,@,#,\$,%,^,&,*]

Здесь соответствует числу 69 в 10-ичной системе исчисления.

В строке сперва идёт запись числа, затем, после символа пробел, на той же строке указана операция которую нужно применить к следующему вводимому с потока ввода числу. После последнего числа операция не будет следовать. После последнего введённого числа должен следовать перевод строки.

Операции над числами разрешается производить только после ввода всех чисел.

Операции будут следующие:

- $a + b$ – складывает числа
- $a - b$ – вычитает числа
- $a * b$ – умножает числа
- a/b – деление чисел
- $a \% b$ – взятие остатка от деления

Программа должна вывести результат последовательного применения данных операций в 4-х системах исчисления (2, 10, 16, из параметров программы), ответ должен быть написан для каждой системы счисления на своей строке в порядке возрастания основания системы счисления. В случае совпадения указанной в параметрах системы счисления с (2, 10, 16) должно быть напечатано 3 строки. При выдаче результата в начале идет система исчисления, далее двоеточие, далее после пробела само число. если номер системы исчисления 2-х цифрный слева должен быть вставлен пробел, чтобы сами выводимые числа начинались с одной и той же позиции.

2 Поиск наибольшего общего делителя

По аналогии с предыдущим заданием в аргументах программы передаются 3 системы исчисления: для числа A , для числа B и для числа C которое является ответом. Длина чисел A и B не более 1000 цифр.

Требуется найти число C – являющееся наибольшим общим делителем чисел A и B . Ответ должен быть записан в 3-ей системе исчисления. Операции необходимо проводить в наибольшей из 3-х систем исчисления.

Для решения можно использовать алгоритм Евклида https://en.wikipedia.org/wiki/Euclidean_algorithm.

3 Возведение матрицы рациональных чисел в степень n

По аналогии с предыдущим заданием в аргументах передаётся 5 чисел:

- *input_notation* – система исчисления исходной матрицы,
- *output_notation* – система исчисления матрицы результата,
- *epsilon* – для вычисления числа точек после запятой,
- *matrix_size* – размерность матрицы
- *degree* – степень в которую необходимо возвести матрицу,

Далее со стандартного потока ввода в матрицу считывается $matrix_size^2$ рациональных чисел. В формате: числитель/знаменатель. Каждая из частей не более 1000 цифр.

Требуется матрицу возвести в степень *degree*, а результат напечатать в стандартный поток вывода в *output_notation*, как числа с плавающей запятой, где для числа знаков после запятой должна обеспечиваться точность до $10^{-epsilon}$.

4 Поиск генов

Требуется написать программу с контролем ошибок. С клавиатуры вводится 2 последовательности. Последовательность РНК: текст в алфавите {A,a,U,u,G,g,C,c,-, пробельный символ}. Признаком конца ввода последовательности служит ввод символа точка. Последовательность белка (аминокислотная последовательность): текст в алфавите представленным 3-им столбцом таблицы 1. Аналогично с последовательностью ДНК, данная последовательность может включать пробельные символы и символ минус.

В таблице 2 представлены правила преобразования РНК в Белок. Требуется вывести координаты начала и конца «гена» кодирующего белок в последовательности (игнорируя пробельные символы и символы минусик). Требуется вывести координаты начала и конца «гена» в тексте (номер строчки, и порядковый номер в строчке). Требуется так же саму последовательность найденного «гена» вывести на экран. Находки должны быть разделены переводом строки, а из печатаемого фрагмента удалены все пробельные символы и символы минусов.

Длина последовательности РНК не превышает 10000 символов, длина последовательности белка не превышает 500 символов.

ввод:

MLII.

```
GCU-AAAAAAAAAA-CU
gcgcgcgcgcgcgcgAUG
CUUAUCAUAUAGAAAAA
AC.
```

вывод:

(2,15) 30 - (3,12) 44: AUGCUUAUCAUAUAG

Таблица 1: Аминокислотные остатки

Аминокислота	сокращ.	1-буква
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Таблица 2: Аминокислотные остатки

Amino acid	Codons
START	AUG
Ala / A	GCU, GCC, GCA, GCG
Arg / R	CGU, CGC, CGA, CGG, AGA, AGG
Asn / N	AAU, AAC
Asp / D	GAU, GAC
Cys / C	UGU, UGC
Gln / Q	CAA, CAG
Glu / E	GAA, GAG
Gly / G	GGU, GGC, GGA, GGG
His / H	CAU, CAC
Ile / I	AUU, AUC, AUA
Leu / L	UUA, UUG, CUU, CUC, CUA, CUG
Lys / K	AAA, AAG
Met / M	AUG
Phe / F	UUU, UUC
Pro / P	CCU, CCC, CCA, CCG
Ser / S	UCU, UCC, UCA, UCG, AGU, AGC
Thr / T	ACU, ACC, ACA, ACG
Trp / W	UGG
Tyr / Y	UAU, UAC
Val / V	GUU, GUC, GUA, GUG
STOP	UAA, UGA, UAG