

## 9 класс (вариант 31991)

### Задача 2

Разработать алгоритм вывода на печать натурального числа, представленного в 32-хзначной системе счисления с помощью букв русского алфавита, от А до Я, исключая Ё, исходное число представлено в двоичной системе, в 64хразрядной сетке 64.

#### Решение:

создание массива символов SA=['А', 'Б', ..., 'Я']

необходимо оценить возможное число разрядов в 32-хзначной системе счисления: достаточно 13 разрядов

Пусть X – исходное число, AX массив из 13 символов для представления X

Для i = 12 до 1

p=Целая часть ( X / 32<sup>i</sup>)

AX [ i+1 ] = SA[p+1]

X=X-p\*32<sup>i</sup>

Конец i

AX[1]=SA[X+1]

Примечание: предполагается, что массив индексируется с 1.

### Задача 4

Разработать алгоритм для проверки правильности расстановки скобок в выражении. Имеется математическое выражение, содержащее скобки, возможно, вложенные, круглые ( ). Например, выражения (A+B и (A+B)) составлены неправильно.

#### Решение:

Задача может решаться построением счетчика открывающихся и закрывающихся скобок:

Open=0

Close=0

Для i=1 до Длина(Expression)

Если Expression[i] == '('

Open=Open+1

Если Expression[i] == ')'

Close=Close+1

Если Close>Open

Вывод ('Ошибка!')

Конец i

Если НЕ(Close==Open)

Вывод ('Ошибка!')

Иное возможное решение задачи связано с применением буфера LIFO (стека). Открывающиеся скобки помещаются в стек, при обнаружении закрывающейся скобки один элемент извлекается из стека. В случае если в какой-либо момент чтение из стека невозможно (стек пуст), а встретилась закрывающаяся скобка – ошибка в выражении. Если стек не пуст по завершении прохода строки – ошибка.

### Задача 5

При обработке звуковых сигналов часто встречается операция нормализации, когда сигнал, представленный в виде массива из  $M$  элементов (целых чисел) меняется так, чтобы он полностью использовал разрядную сетку, например, был представлен неотрицательным целым числом, 8 разрядов в двоичном коде, при этом не должно возникать ситуации, когда старший и младший разряд не меняются на протяжении всего сигнала. Необходимо сформулировать алгоритм, такой, чтобы выполнить нормализацию, если входной сигнал задан, как массив целых чисел со знаком, 16 разрядов.

**Решение:**

Задача решается в два действия:

- находятся максимальный и минимальный элементы массива  $M$

$Max = -32768$

$Min = 32767$

Для  $i = 1$  до Длина( $M$ )

Если  $M[i] > Max$

$Max = M[i]$

Если  $M[i] < Min$

$Min = M[i]$

Конец  $i$

- собственно, нормализация:

Если  $Max == Min$

Вывод (“Вырожденный массив, все элементы равны между собой!”)

Иначе

$Scale = 255 / (Max - Min)$

Для  $i = 1$  до Длина( $M$ )

$N[i] = \text{Целое}((M[i] - Min) * Scale)$

Конец  $i$

***9 класс (вариант 32991)***

**Задача 1**

В организации построена большая компьютерная сеть, имеющая древовидную топологию – к главному концентратору присоединяются вспомогательные концентраторы, к ним, в свою очередь – рабочие станции и новые концентраторы и так далее. Однако, есть подозрение, что один

из сотрудников добавил дополнительное соединение между двумя концентраторами, образовав замкнутый контур. Помогите администратору сети определить, есть ли такое соединение, составив соответствующий алгоритм. Примечание: у каждого концентратора и у каждой рабочей станции свой уникальный адрес, доступна функция запроса адресов, подключенных к концентратору, доступна функция «установить флажок» - установить на концентраторе 0 или 1 значение логической переменной.

### **Решение:**

Поскольку заранее неизвестно, какое количество уровней разветвления имеет сеть, целесообразно использовать рекурсивно вызываемую функцию, производящую опрос для каждого концентратора и устанавливающую флажок на пройденных концентраторах:

Функция Опрос(Адрес): логическая

Loop=Получить флажок на устройстве

Если Loop=Истина

    Вывести(“Найдено соединение, образующее контур”)

    Прекращение программы

Иначе

    Установить флажок на устройстве (Истина)

    Адреса = Получить адреса подключенных устройств

    Пока  $i < \text{Длина(Адреса)}$

        Если Тип устройства(Адреса[i]) == ‘Концентратор’

        Loop=Опрос(Адреса[i])

    Конец  $i=i+1$

    Возврат Loop

Как только будет найдено повторение «поднятого» флажка, делаем вывод о наличии замкнутого контура в сети – так как мы прошли двумя маршрутами через один и тот же концентратор.

### **Задача 5**

При обработке медицинских диагностических сигналов часто встречается операция центрирования, когда последовательность чисел смещается (к каждому элементу добавляется некоторое число) так, чтобы среднее значение всех элементов последовательности было равно 0. Выполните центрирование последовательности чисел, представленных в виде массива.

### **Решение:**

Задача решается в два действия:

1) определяется среднее значение сигнала:

Average=0.0

Для  $i=1$  до КоличествоЧисел

$Average = Average + A[i]$

Конец  $i=i+1$

$Average = Average / \text{КоличествоЧисел}$

2) Каждое число последовательности смещается на -Среднее:

Для  $i=1$  до КоличествоЧисел

$A[i] = A[i] - Average$

Конец  $i=i+1$

*10 класс (вариант 31101)*

**Задача 3**

При анализе астрофизических данных в некоторых случаях целесообразно выяснить, вокруг каких объектов формируются компактные группы — скопления объектов, лежащих на относительно небольшом расстоянии. Будем считать, что расстояние между центральным и периферийными объектами в компактной группе меньше, чем среднее расстояние между объектами парами ближайших друг к другу объектов  $M$ . Предложите алгоритм поиска центров таких компактных групп. Входные данные: массив  $[2, N]$  содержащий  $N$  пар координат  $(x, y)$  объектов.

**Решение:**

- формирование матрицы расстояний между объектами

$A=[2, N]$

$D = [N, N]$

Для  $i = 1$  до  $N$

    Для  $j = 1$  до  $N$

        Если  $i=j$

$D[i, j]=0$

        Иначе

$D[i, j]=(A[1, i]-A[1, j])^2 + (A[2, i]-A[2, j])^2$

    Конец  $j$

Конец  $i$

- для каждого объекта надо найти ближайший и сформировать массив минимальных расстояний

$Dm=[N]$

Для  $i=1$  до  $N$

$min=D[i, 1]$

    Для  $j=2$  до  $N$

        Если  $i \neq j$

            Если  $D[i, j] < min$

$min=D[i, j]$

    Конец  $j$

$Dm[i]=min$

Конец  $i$

- необходимо рассчитать среднее расстояние между соседними объектами:

$$M=0$$

Для  $i=1$  до  $N$

$$M=M+Dm[i]$$

Конец  $i$

$$M=M/N$$

- необходимо определить, сколько объектов лежат ближе к данному, чем на среднем расстоянии  $M$ :

$$Nobj=\text{Целое}[N]$$

Для  $i=1$  до  $N$

$$Nobj[i]=0$$

Для  $j=1$  до  $N$

Если  $D[i,j]<M$

$$Nobj[i]=Nobj[i]+1$$

Конец  $j$

Конец  $i$

- объект с номером  $i$  рассматривается как центр группы из  $Nobj[i]$  объектов (включая себя самого), скоплением можно именовать группу из трех и более объектов.

## Задача 5

При обработке медицинских изображений часто встречается операция нормализации, когда изображение, представленное в виде матрицы из  $M \times N$  элементов (целых чисел) меняется так, чтобы оно полностью использовало разрядную сетку, например, было представлено неотрицательным целым числом, 8 разрядов в двоичном коде, при этом не должно возникать ситуации, когда старший и младший разряд не меняются на протяжении всего сигнала. Необходимо сформулировать алгоритм, такой, чтобы выполнить нормализацию, если входной сигнал задан, как матрица целых чисел со знаком, 16 разрядов.

### Решение:

Задача решается в два действия:

- находятся максимальный и минимальный элементы матрицы  $A$

$$Max=-32768$$

$$Min=32767$$

Для  $i = 1$  до  $M$

Для  $j = 1$  до  $N$

Если  $A[i,j] > \text{Max}$

$\text{Max} = A[i,j]$

Если  $A[i,j] < \text{Min}$

$\text{Min} = A[i,j]$

Конец  $j$

Конец  $i$

- собственно, нормализация:

Если  $\text{Max} == \text{Min}$

Вывод (“Вырожденная матрица, все элементы равны между собой!”)

Иначе

$\text{Scale} = 255 / (\text{Max} - \text{Min})$

Для  $i = 1$  до  $M$

Для  $j = 1$  до  $N$

$A_{\text{norm}}[i,j] = A[i,j] * \text{Scale}$

Конец  $j$

Конец  $i$

## 10 класс (вариант 32101)

### Задача 2

Различают префиксную, инфиксную и постфиксную запись математических выражений: пре», «пост» и «ин» относятся к относительной позиции оператора по отношению к обоим операндам. В префиксной записи операция предшествует обоим операндам, в постфиксной записи операция следует за двумя операндами, а в инфиксной записи операция разделяет два операнда.

$A+B$  – инфиксная запись

$+A B$  – префиксная запись

$A B +$  постфиксная запись

Операции: сложение (+), вычитание (-), умножение (\*), деление (/) и возведение в степень (^). Для этих операций установлен следующий порядок вычислений (от высшего к низшему): возведение в степень, умножение/деление, сложение/вычитание. Этот порядок можно изменить при помощи скобок.

Составить алгоритм преобразования выражения в постфиксную форму.

$$a *(b+ c) / e ^ f ^ d - g * h + k$$

Решение:

Следует анализировать символы входной последовательности, различая операнды (a,b,c,...) и операторы. Будем формировать список операндов и операторов на выходе, применяя для установки приоритета операций буфер LIFO (стек), куда будем предварительно помещать операторы:

Если символ является операндом, то добавить его в конец выходного списка.

Если символ является левой скобкой, положить его в стек.

Если символ является правой скобкой, то выталкивать элементы из стека, пока не будет найдена соответствующая левая скобка. Каждый оператор добавлять в конец выходного списка.

Если символ является оператором \*, /, + или -, поместить его в стек. Однако, перед этим вытолкнуть любой из операторов, уже находящихся в стеке, если он имеет больший или равный приоритет, и добавить его в результирующий список.

Когда входное выражение будет полностью обработано, проверить стек. Любые операторы, всё ещё находящиеся в нём, следует вытолкнуть и добавить в конец итогового списка.

Для приведенного примера:

$$a *(b+ c) / e ^ f ^ d - g * h + k$$

Шаг	Выходной список	Стек
1	a	
2	a	*
3	a	*, (
4	a	*, (
5	a, b	*, (
6	a, b	*, (, +
7	a, b, c	*, (, +
8	a, b, c, +	*
9	a, b, c, +, *	/
10	a, b, c, +, *, e	/
11	a, b, c, +, *, e,	/, ^



12	a, b, c, +, *, e, f	/, ^
13	a, b, c, +, *, e, f, ^	/, ^
14	a, b, c, +, *, e, f, ^, d	/, ^
15	a, b, c, +, *, e, f, ^, d, ^, /	-
16	a, b, c, +, *, e, f, ^, d, ^, /, g	-
17	a, b, c, +, *, e, f, ^, d, ^, /, g	-, *
18	a, b, c, +, *, e, f, ^, d, ^, /, g, h	-, *
19	a, b, c, +, *, e, f, ^, d, ^, /, g, h, *, -	+
20	a, b, c, +, *, e, f, ^, d, ^, /, g, h, *, -, k	+
21	a, b, c, +, *, e, f, ^, d, ^, /, g, h, *, -, k, +	

Ответ:  $abc+*ef^d^/gh*-k+$

### Задача 5

Ученые исследуют сложную динамическую (меняющуюся во времени) систему. Состояние системы представляется множеством точек с координатами (x,y) и в каждой точке определяется вектор  $v=(v_x, v_y)$ . Выяснилось, что более половины векторов являются коллинеарными и есть подозрение, что не менее 90% остальных векторов – отклонены на одинаковый угол относительно них. Предложите алгоритм проверки этой гипотезы.

#### Решение:

Сначала проверим первую часть гипотезы:

так как вектора расположены на плоскости (имеют только две компоненты), то у коллинеарных векторов модуль скалярного произведения равен произведению длин (модулей) самих векторов:

$$|v_x[i]*v_x[j]+v_y[i]*v_y[j]| == \sqrt{v_x[i]^2+v_y[i]^2}*\sqrt{v_x[j]^2+v_y[j]^2}$$

Перебирая индекс i и для каждого i – все значения j, кроме i=j, необходимо фиксировать число коллинеарных пар. Если, перебрав половину возможных значений индекса i, мы получили наибольшее число коллинеарных пар меньше половины общего числа векторов N – гипотеза не подтверждена, более половины векторов – не коллинеарны. Целесообразно создать массив размером (N) содержащий значение косинуса угла между векторами:

$$M[j] = (v_x[i] * v_x[j] + v_y[i] * v_y[j]) / (\sqrt{v_x[i] * v_x[i] + v_y[i] * v_y[i]} * \sqrt{v_x[j] * v_x[j] + v_y[j] * v_y[j]})$$

заполнив его значениями для такого номера  $i$ , для которого число коллинеарных пар оказалось больше  $N/2$ . Пусть массив содержит  $n$  единиц, соответствующих коллинеарным парам. Тогда, если массив содержит больше, чем  $0,9 * n$  одинаковых значений – гипотеза может считаться доказанной.

Подсчет одинаковых значений можно осуществлять, например так:

$C = \text{Ложь}$

$i = 1$

Пока  $(\text{НЕ}(C)) \text{ И } (i \leq N)$

$m = 0$

Если  $\text{НЕ}(M[i] == 1)$

Для  $j = 1$  до  $N$

Если  $M[i] == M[j]$

$m = m + 1$

Конец  $j = j + 1$

Если  $m \geq 0,9 * n$

$C = \text{Истина}$

$i = i + 1$

Конец Пока

Если  $C == \text{Истина}$

Вывод (“Гипотеза подтверждена!”)

## 11 класс (вариант 31111)

### Задача 4

Растровое изображение представляется в виде двумерного массива (матрицы) точек — пикселей. Один из методов изменения разрешения изображения предполагает отбрасывание половины пикселей, но так, чтобы не отбрасывались соседние сверху, снизу, слева и справа пиксели. Предложите алгоритм реализации такого метода и формирования результирующего изображения.

#### Решение:

Пусть  $M=[m,n]$  - двумерный массив (матрица) пикселей на входе. Для реализации заданного преобразования достаточно применить «шахматную» схему исключения пикселей. В свою очередь, для этого достаточно исключать каждый четный пиксель в четных строках и каждый нечетный пиксель - в нечетных строках. Однако, необходимо учесть, что число столбцов  $n$  может быть нечетным. В этом случае для получения выходного изображения придется уменьшить на 1 диапазон изменения столбцов.

$n\_new = \text{Целое}(n / 2)$

$M\_new = [m, n\_new]$

Для  $i=1$  до  $m$

    Для  $j=1$  до  $n\_new$

        Если  $\text{Остаток}(i/2) = 0$

$M\_new[i,j] = M[i, 2*j-1]$

        Иначе

$M\_new[i,j] = M[i, 2*j]$

    Конец  $j$

Конец  $i$

Верным является решение с изменением числа строк и сохранением числа столбцов.

### Задача 5

Для обработки мультимедийных сигналов (аудио, изображения, видео) часто применяют различные математические преобразования, в частности, основанные на вычислении тригонометрических функций. Точность при этом бывает не очень важна, гораздо важнее скорость вычислений. Предложите способ приближенного вычисления синуса произвольного угла (задается в градусах от 0 до 360) с использованием не более одной операции деления, одной - умножения и двух-трех сложений и/или вычитаний, в памяти ЭВМ можно хранить не более 5 хорошо известных значений синуса для 0, 30, 45, 60 и 90 градусов. Можно использовать условные операторы.

#### Решение:

Для минимизации числа трудоемких операций при вычислении сложных функций применяют линейную интерполяцию, сохраняя в памяти табличные значения функции (например, в массиве  $A=[0, 0.5, 0.71, 0.87, 1]$ ). Для промежуточных значений угла строится линейная пропорция.

Для вычисления синуса промежуточного угла  $\phi$ :

Если  $(\phi \geq 0)$  И  $(\phi < 30)$

$$\sin = (A[2] - A[1]) * \phi / 30$$

Если  $(\phi \geq 30)$  И  $(\phi < 45)$

$$\sin = (A[3] - A[2]) * (\phi - 30) / 15 + A[2]$$

Если  $(\phi \geq 45)$  И  $(\phi < 60)$

$$\sin = (A[4] - A[3]) * (\phi - 45) / 15 + A[3]$$

Если  $(\phi \geq 60)$  И  $(\phi \leq 90)$

$$\sin = (A[5] - A[4]) * (\phi - 60) / 30 + A[4]$$

Для вычисления синуса углов, больших 90 градусов следует воспользоваться нечетностью и периодичностью функции. Возможно исключение операции деления (путем замены на соответствующий множитель).

## 11 класс (вариант 32111)

### Задача 2

Различают префиксную, инфиксную и постфиксную запись математических выражений: пре-, «пост» и «ин» относятся к относительной позиции оператора по отношению к обоим операндам. В префиксной записи операция предшествует обоим операндам, в постфиксной записи операция следует за двумя операндами, а в инфиксной записи операция разделяет два операнда.

$A+B$  – инфиксная запись

$+A B$  – префиксная запись

$A B +$  постфиксная запись

Операции: сложение (+), вычитание (-), умножение (\*), деление (/) и возведение в степень (^). Для этих операций установлен следующий порядок вычислений (от высшего к низшему): возведение в степень, умножение/деление, сложение/вычитание. Этот порядок можно изменить при помощи скобок.

Составить алгоритм вычисления выражение, записанного в постфиксной форме (числа однозначные), вычислить:

$$3\ 5\ 1 + 2\ 4 * - 1\ 2\ 6 + ^ / +$$

Решение:

Для анализа постфиксной записи удобно использовать буфер LIFO (или стек) — строка считывается слева направо, формируя стек операндов. Каждый раз, когда встречается знак операции, из стека извлекаются два (последних) операнда и производится вычисление результата, который, в свою очередь помещается в стек.

Для приведенного примера:

- 1) стек: 3
- 2) стек: 3, 5
- 3) стек: 3, 5, 1
- 4) стек: 3, 6 (считали «+» и выполнили чтение двух операндов из стека 5+1, результат — в стек)
- 5) стек 3, 6, 2
- 6) стек 3, 6, 2, 4
- 7) стек 3, 6, 8
- 8) стек 3, -2
- 9) стек 3, -2, 1
- 10) стек 3, -2, 1, 2
- 11) стек 3, -2, 1, 2, 6
- 12) стек 3, -2, 1, 8
- 13) стек 3, -2, 1
- 14) стек 3, -2
- 15) стек 1

Ответ: 1

#### Задача 4

Каждому символу алфавита (кроме Ё и Й) поставлено в соответствие число (см. таблицу):

	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Для шифрования сообщения цифры в таблице смещаются на определенное число (напр. «а»-6, «м»-6 или «в»-8, «а»-8, «у»-8) и затем выписывается полученное сообщение. Так, со смещением в 1 символ, слово ПРИВЕТ превратится в РСКГЖУ. Известно, что перехваченное сообщение (задано массивом символов) зашифровано таким способом и содержит слово «алекс». Предложите алгоритм расшифровки, эффективность приветствуется!

#### Решение

Конечно, формально правильным решением будет перебор всех вариантов сдвига и попытка расшифровка сообщения с поиском подстроки «алекс» в расшифрованном тексте. Однако более эффективным способом является вычисление разности между кодами букв слова «алекс»: 10,-5,4,7 – эта разность сохраняется при указанном способе кодирования. Следовательно, можно

вычислить разность кодов соседних букв зашифрованного текста, а затем найти в полученном массиве чисел последовательность 10,-5,4,7 . Эта последовательность следует за буквой «а» зашифрованного текста. Таким образом, сдвиг найден, а текст – расшифрован.