ЗАДАНИЕ ПО ИНФОРМАТИКЕ

ВАРИАНТ 37101 для 10-го класса

1. Кластеризация данных – можно ли разделить массив n точек на плоскости на m частей так, чтобы расстояние всех элементов от среднего в пределах части (кластера) было меньше половины расстояния между средними значениями любой пары частей? Предложите алгоритм проверки для заданного массива точек, m и n.

<u>Ответ:</u> возможны варианты решения. Целесообразным представляется выделить функцию, возвращающую расстояние между точками: R[i,j]=sqrt((x[i]-x[j])^2+ (y[i]-y[j])^2) и сформировать группы (кластеры) точек, по p[i] = Целая часть (n / m) элементов в каждом, добавив оставшиеся точки в последний кластер, число элементов в нем p[m]=p[m-1] + Остаток (n/m). Первоначальное распределение точек по кластерам может быть произвольным.

Средняя точка в пределах кластера может быть вычислена как:

k=1

Цикл по i=1 до m (по кластерам)

M[i]=0

Цикл по j=1 до p[i] (по элементам кластера)

XM [i]=XM[i]+x[k]

YM [i]=YM[i]+y[k]

k=k+1

Конец j=j+1

XM[i]=XM[i]/p[i] (среднее значение для кластера)

YM[i]=YM[i]/p[i] (среднее значение для кластера)

Конец i=i+1

Проверка условия при этом выглядит так:

k=1

Condition = Истина

Cmin= sqrt((XM[1]-XM[m])^2+ (XM[1]-YM[n])^2)

Цикл по i=1 до m (по кластерам)

Цикл по j=1 до m (по кластерам)

C=Cmin

Если i~=j то C= sqrt((XM[i]-XM[j])^2+ (YM[i]-YM[j])^2)

Если C<Cmin то Cmin=C

Цикл по i=1 до m (по кластерам)

Dmax=0

Цикл по j=1 до p[i] (по элементам кластера)

D= sqrt((XM[i]-x[k])^2+ (YM[i]-y[k])^2)

Если D>Dmax то Dmax=D

k=k+1

Конец j=j+1

Если Cmin<2*Dmax то Condition = Ложь (проверка условия для каждого кластера)

Конец i=i+1

Если проверка дала неудовлетворительный результат можно перераспределить элементы между кластерами и повторять проверку условия и перераспределение элементов до тех пор, пока не образуются вырожденные (пустые кластеры). При этом условие исключения точки из кластера – максимальное расстояние от точке до центра кластера, условие добавление в кластер – минимизация расстояния до центра кластера.

2. Классификация по методу опорных векторов – построить алгоритм проверки принадлежности заданной точки (x,y,z) в пространстве классу – нижнему или верхнему полупространству относительно плоскости-классификатора. Обученный классификатор: d = ax+by+cz

<u>Ответ:</u> достаточно проверить условие

Low = Истина (принадлежность ниже границы)

Если z > (d-a*x-b*y)/c то Low = Ложь

High = Истина (принадлежность нижней полуплоскости)

Если z < (d-a*x-b*y)/c то High = Ложь

Соответственно, при строгом выполнении равенства, нельзя вынести определенного суждения о принадлежности точки определенному классу

3. Предложите алгоритм суммирования двух двухразрядных чисел с использованием только логических функций И, НЕ

<u>Ответ:</u>

Пусть A0, A1 – младший и старший разряды первого операнда, B0, B1 – младший и старший разряды второго операнда, S0, S1, S2 – разряды суммы

Тогда S0 = (A0 И (НЕ B0)) ИЛИ (B0 И (НЕ A0)) = НЕ (НЕ(A0 И (НЕ B0)) И НЕ(B0 И (НЕ A0)))

перенос C = A0 И B0

S1 = (A1 И (НЕ (B1) И (НЕ (C)) *ИЛИ* (B1 И (НЕ (A1) И (НЕ (C)) *ИЛИ* (C И (НЕ (B1) И (НЕ (A1)) *ИЛИ* (A1 И B1 И C)

S2 = (A1 И B1) ИЛИ (A1 И C) ИЛИ (B1 И C)

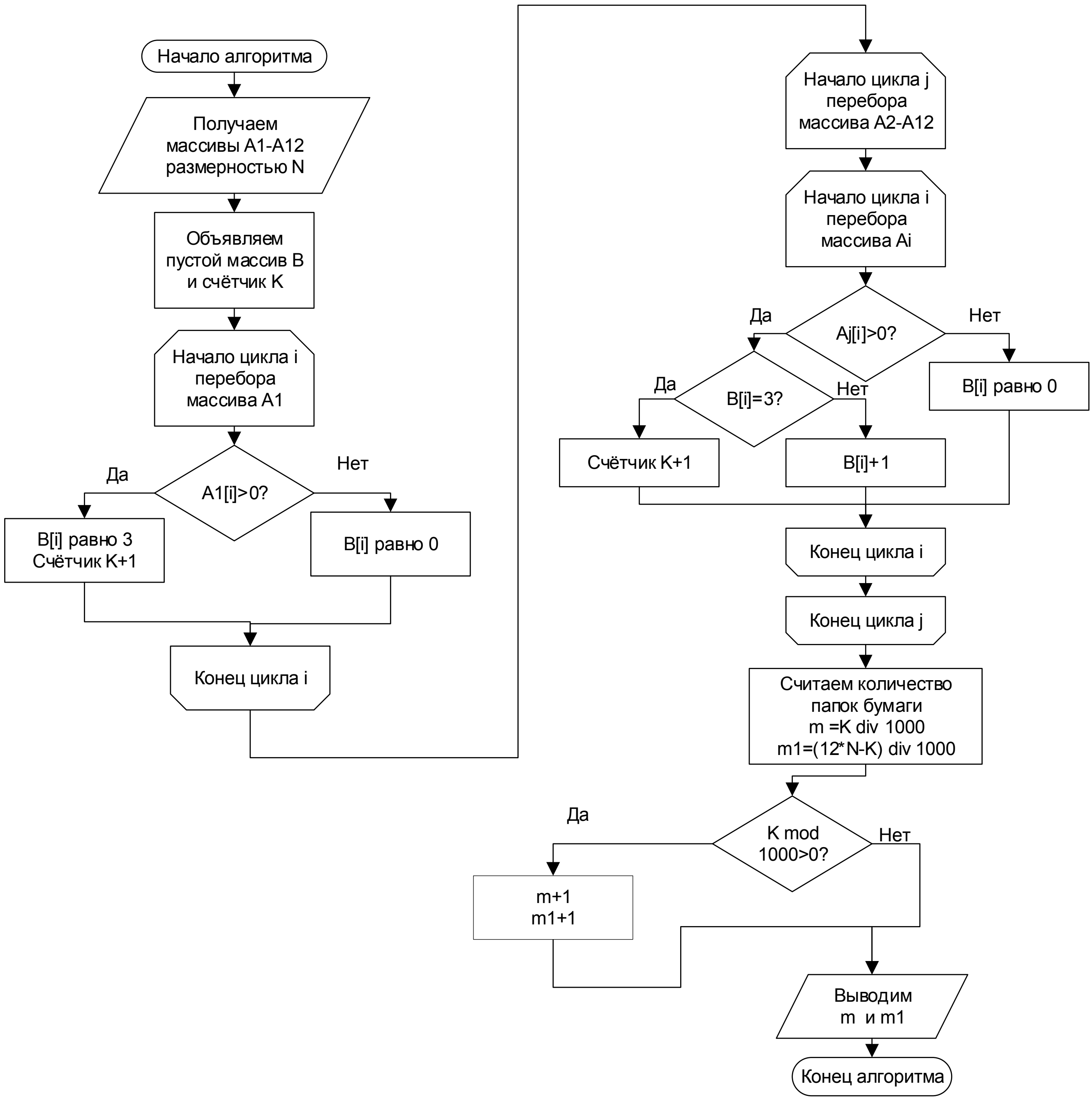
причем каждая операция A *ИЛИ* B заменяется на НЕ (НЕ(A) И НЕ(B))

4. Для суммирования модулированной последовательности чисел без потери точности предлагается хранить результат в 256-разрядном числе S. Какова длина последовательности целых положительных 8-разрядных чисел, такая, что сумма сможет быть размещена в S, если модуляция последовательности осуществляется перемножением элемента на случайно выбранное число из массива [1, 3, 8].

Ответ: предполагая двоичные разряды, получаем наибольшее число, равное 2^{256}-1. В худшем случае каждое число в последовательности имеет 8 разрядов, равных 1, то есть 2^{8}-1, умножается на 8 (т.е. сдвигается на три двоичных разряда). Число элементов последовательности будет равно отношению 2^{253}-1 к 2^{8}-1.

5. Управляющей компании необходимо ежемесячно печатать единые платёжные документы, уведомляющих потребителей о сумме оплаты электроэнергии. Количество таких документов N не менее 25000. При этом для лиц, имеющих задолженность ($Q>0$) за электричество более 3 месяцев, документ печатается на листе красного цвета. В конце каждого года компания оформляет заказ на покупку белой бумаги и красной бумаги упаковками по 1000 листов, исходя из количества потраченной в прошедшем году, игнорируя неизрасходованный остаток.

Печать данных о потреблённой и оплаченной электроэнергии осуществляется в соответствии с базой данных компании, где для каждого из N потребителей указана величина его задолженности или нулевое значение при её отсутствии. Представьте в виде блок-схемы алгоритм работы программы, вычисляющей расход упаковок бумаги за 12 месяцев прошедшего года, если известно, что в январе компания из-за социально-экономической ситуации решила нарушить собственные правила и причислить к должникам всех лиц, имеющих задолженность.


Начало алгоритма
Получаем массивы A1-A12 размерностью N
Объявляем пустой массив B и счётчик K
Начало цикла i перебора массива A1
A1[i]>0?
Да
Нет
B[i] равно 3 Счётчик K+1
B[i] равно 0
Конец цикла i
Начало цикла j перебора массива A2-A12
Начало цикла i перебора массива Ai
Aj[i]>0?
Да
Нет
B[i]=3?
Да
Нет
B[i] равно 0
Счётчик K+1
B[i]+1
Конец цикла i
Конец цикла j
Считаем количество папок бумаги m =K div 1000 m1=(12*N-K) div 1000
K mod 1000>0?
Да
Нет
m+1 m1+1
Выводим m и m1
Конец алгоритма