

ЗАДАНИЕ ПО ИНФОРМАТИКЕ
ВАРИАНТ 32101 для 10-го класса

Разрабатывать алгоритмы необходимо на языке блок-схем, псевдокоде или естественном языке

1. Предложите алгоритм, как установить, что заданные четыре точки (координаты точек заданы парами чисел с плавающей точкой) являются вершинами прямоугольника.

Решение: Одним из способов является вычисление расстояний между парами точек по формуле $d_{ij} = \sqrt{(x[i]-x[j])^2 + (y[i]-y[j])^2}$, всего 6 сочетаний, и проверка, выполняется ли в совокупности следующий набор условий – три пары расстояний оказываются равными друг другу, причем два – строго больше остальных четырех (последняя проверка позволяет исключить вырожденный случай, когда все шесть расстояний равны нулю).

2. Микроконтроллер должен вычислить сумму произведений тридцати трех пар 12-разрядных натуральных чисел. Возможно ли провести вычисления, используя 32-битные регистры для хранения операндов и результата?

Решение: произведение $2^{12} * 2^{12}$ требует $2^{(12+12)} = 2^{24}$ – 24 разрядов. Суммирование 33 пар может добавить еще 6 разрядов, таким образом, требуется 30 разрядов, что меньше, чем 32. Следовательно, ответ – да, возможно.

3. Необходимо оценить успеваемость учеников школы по триместровым оценкам. Предложите алгоритм выявления учеников, чей балл в первом триместре был ниже, а во втором и третьем - стал выше медианного балла (такого, чтобы половина учеников имела балл выше медианного).

Решение (вариант): Пусть N – число учеников, массив из 11 элементов $p[i]$ – число предметов (оценок) за триместр для 11 классов. В массиве $A[i,j,k]$ хранятся оценки учеников по триместрам (размер массива $N \times n \times 3$, где $n = \max(p[i])$, при этом предполагаем, что массив A предзаполнен нулями). Достаточно рассчитать средний балл каждого ученика и затем отсортировать массив средних баллов по возрастанию (убыванию) за каждый триместр. В середине отсортированного массива мы обнаружим медианный балл.

Ввод N, n, A

Average = массив[N,3]

Для $p = 1$ до 3

 Для $i = 1$ до N

$k = 0$

$S = 0$

 Для $j = 1$ до n

$S = S + A[i,j,p]$

 Если $A[i,j,p] \neq 0$ То $k = k + 1$

 Конец

 Average[i,p] = S/k

 Конец

$S = 0$

 Для $j = 1$ До $N - 1$

 Для $i = 0$ До $N - 1 - j$

 Если Average[i,p] > Average[i+1,p] То

$S = \text{Average}[i,p]$

 Average[i,p] = Average[i+1,p]

 Average[i+1,p] = S

 Конец

 Конец

Олимпиада школьников «Надежда энергетики». Отборочный этап. Заочная форма.

```
Конец
u[p]=0
Если Остаток(N/2)==0 То
    u[p]=0.5*(Average[Целая часть(N/2)+1,p]+Average[Целая часть(N/2),p])
Иначе u=Average[Целая часть(N/2)+1,p]
Конец
Для i = 1 до N
    Если (Average[i,1]<u[1])И(Average[i,2]>u[2])И(Average[i,3]>u[3]) То
        Вывод(«Найден подходящий ученик!»)
Конец
```

4. При анализе трафика и оптимизации маршрутизации в сетях необходимо убедиться в отсутствии повторения узлов в сети. Предложите алгоритм поиска повторяющихся узлов, если каждое устройство имеет список устройств, соединенных непосредственно с ним, для опроса устройства необходимо иметь доступ к нему (построить к нему маршрут от исходного устройства).

Решение (вариант):

Необходимо в цикле пройти по устройствам, составляя общий список пройденных устройств. Добавляя каждый новый элемент списка следует проверять, присутствует ли данное устройство в списке. В случае, если возникает дублирование устройства в списке пройденных, фиксируется факт наличия замкнутого контура.

5. Предложите алгоритм генерации случайной таблицы из 8x8 символов (128 вариантов символов с кодами 0-127) такой, чтобы гарантировать отсутствие повторения символов. При разработке считайте, что необходимо минимально возможное число раз использовать генератор случайных чисел (выдает случайное число с плавающей точкой в диапазоне 0 - 1).

Решение (вариант):

Поскольку необходимы 64 случайных символа, минимальное число вызовов генератора случайных чисел равно 64. Для исключения повторов следует при выборе очередного символа исключать его из выбора в дальнейшем:

Пусть коды символов находятся в массиве A[i]

В=массив[8,8]

n=128

Для i=1 До 64

k = Целая часть(random()*(n-1))+1

В[Целая часть((i-1)/8)+1, Остаток((i-1)/8)+1]= A[k]

S=A[k]

A[k]=A[n]

A[n]=S

n=n-1

Конец