**Обзор заданий КЕГЭ по информатике и краткие рекомендации по их решению**

|  |
| --- |
| 1 Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы) – *задание со схемой дорог и таблицей* |
| 2 Умение строить таблицы истинности и логические схемы  *Для построения таблицы истинности лучше написать небольшую программу на Питон*  for x in range (2):  for y in range (2):  for w in range (2):  for z in range (2):  if (((not w or not x) == (not z or y)) and (y or w)):  print (x,y,w,z)   |  |  |  |  |  | | --- | --- | --- | --- | --- | | x Λ y | x V y | x → y | x ≡ y | ¬x | | x and y | x or y | not x or y  или  x<=y | x == y | not x |   Очень важно:   * если перед not стоит операция сравнения, необходимо выражение с not взять в скобки * если всё выражение истинно, можно не писать в конце «==1:» * если всё выражение ложно, необходимо взять его в скобки и в конце обязательно написать «==0:» |
| 3 Знание о технологии хранения, поиска и сортировки информации в реляционных базах данных –  *задание на поиск родственников: бабушек, дядей и т.д.* |
| 4 Умение кодировать и декодировать информацию. *Как правило, задача на условие Фано и лучше построить двоичное дерево* |
| 5 Формальное исполнение алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд. Если алгоритм предполагает перевод чисел из десятичной системы в двоичную и из двоичной в десятичную, то удобно воспользоваться функциями Excel (категория «Инженерные»): ДВ.В.ДЕС() и ДЕС.В.ДВ() |
| 6 Знание основных конструкций языка программирования, понятия переменной, оператора присваивания  *Для решения этой задачи можно устроить перебор значений (главное определиться с интервалом перебора) Задание из Демо-2020:* *Определите, при каком наименьшем введённом значении переменной s программа выведет число 64.*   |  |  | | --- | --- | | Программа:  s = int(input())  n = 1  while s < 51:  s = s + 5  n = n \* 2  print(n) | Переборное решение:  for x in range(10,200):  s = x  n = 1  while s < 51:  s = s + 5  n = n \* 2  if n==64:  print(x, n) | |
| 7 Умение определять объём памяти, необходимый для хранения графической и звуковой информации. ***Важно*** *знать главную формулу информатики 2i=N* |
| 8 Знание о методах измерения количества информации. *Задачи на комбинаторику или кодирование слов по алфавиту, которые решаются с помощью систем счисления.*  Задачи на комбинаторику можно решить также с помощью кода. Например, такое условие: Вася составляет 6-буквенные слова, в которых есть только буквы К, А, Н, Т, причём буква К используется в каждом слове ровно 2 раза. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?  Решение: s='KAHT'  k=0  for a1 in s:  for a2 in s:  for a3 in s:  for a4 in s:  for a5 in s:  for a6 in s:  s1=a1+a2+a3+a4+a5+a6  if s1.count('K')==2:  k+=1  print(k) |
| 9 Умение обрабатывать числовую информацию в электронных таблицах  *Надо знать основные функции ЭТ: МИН, МАКС, СРЗНАЧ, СУММ и т.д* |
| 10 Информационный поиск средствами операционной системы или текстового процессора  *Может пригодиться задать Найти -> Параметры -> Только слова целиком* |
| 11 Умение подсчитывать информационный объём сообщения  ***Важно*** *знать главную формулу информатики 2i=N* |
| 12 Умение анализировать результат исполнения алгоритма  *Если в задаче используют Исполнитель Редактор, то можно написать программку на Питон*  В этом примере известно какую строчку подали на вход и надо определить, что получится после выполнения программы  s='>'+20\*'0'+30\*'1'  while '>1' in s or '>0' in s:  if '>10' in s:  s=s.replace('>10','3>',1)  elif '>0' in s:  s=s.replace('>0','2>',1)  if '>1' in s:  s=s.replace('>1','0>0',1)  print(s)  print(s.count('2'))  ***Важно*** *в методе replace задать третий параметр – 1, поскольку ищется первое вхождение подстроки в строке.*  В некоторых задачах, известно какая строка будет получена на выходе и надо узнать, что подавали на вход. Например, *Дана программа для редактора:*  НАЧАЛО  ПОКА нашлось (01) ИЛИ нашлось (02) ИЛИ нашлось (03)  заменить (01, 2302)  заменить (02, 10)  заменить (03, 201)  КОНЕЦ ПОКА  КОНЕЦ  *Известно, что исходная строка начиналась с нуля, а далее содержала только единицы, двойки и тройки. После выполнения данной программы получилась строка, содержащая 40 единиц, 10 двоек и 8 троек. Сколько единиц было в исходной строке?*  for a in range(100):  for b in range(100):  for c in range(100):  s='0'+a\*'1'+b\*'2'+c\*'3'  while '01' in s or '02' in s or '03' in s:  s=s.replace('01','2302',1)  s=s.replace('02','10',1)  s=s.replace('03','201',1)  if s.count('1')==40 and s.count('2')==10 and s.count('3')==8:  print(a) |
| 13 Умение представлять и считывать данные в разных типах информационных моделей (схемы,  карты, таблицы, графики и формулы). *Задача на подсчёт количества дорог из одного пункта в другой* |
| 14 Знание позиционных систем счисления  ***Важно*** *знать формулы*  число aN в системе счисления с основанием a записывается как единица и N нулей:  число aN – aM записывается в системе счисления с основанием a как *N-M* старших цифр этой системы счисления, за которыми стоят M нулей:  Можно написать программу на Питон. Например, такое условие. *Значение выражения 3435 – 79 + 48 записали в системе счисления с основанием 7. Сколько цифр 6 содержится в этой записи?*   |  |  | | --- | --- | | **Способ 1**  r=343\*\*5-7\*\*9+48  s=''  while r>0:  d=r%7  s=s+str(d)  r=r//7  print(s.count('6')) | **Способ 2**  r=343\*\*5-7\*\*9+48  k=0  while r>0:  if r%7==6:  k+=1  r=r//7  print(k) | |
| 15 Знание основных понятий и законов математической логики  ***Важно*** *знать законы алгебры логики, таблицы истинности логических операций, законы де Моргана и формулу* А→В = not A + B  Эти задачи бывают разных видов. Рассмотрим некоторые из них.  # Amax-? **Для какого числа А?**  # ((x <= 9) -> (x\*x <= A)) /\ ((y\*y <= A) -> (y <= 9))=1  for A in range(200):  k=0  for x in range(100):  for y in range(100):  if (x>9 or x\*x<=A) and (y\*y>A or y<=9):  k+=1  if k==10000:  print(A)    # Ответ: 99    # Amin-? Amax-? **Конъюнкции**  # ( x & 25 <> 1) V ((x & 34 = 2) -> (x & A = 0))=1  for A in range(1,500):  k=0  for x in range(1000):  if x&25!=1 or x&34!=2 or x&A==0:  k+=1  if k==1000:  print(A)    # Ответ: 8 56    # Amax-? **Делители**  # Дел(x,18)->(не Дел(x,A) -> не Дел(x,12))=1  for A in range(1,200): # ВАЖНО от 1, т.к. на 0 делить нельзя!  k=0  for x in range(100):  if x%18!=0 or x%A==0 or x%12!=0:  k+=1  if k==100:  print(A)  # Ответ: 36  Можно вместо счётчика k использовать флажок, который изначально делаем равным 0. Если выражение окажется ложным хоть один раз, присваиваем флажку единицу. Предыдущая задача без счётчика:  for A in range(1,200):  k=0  for x in range(100):  if (x%18!=0 or x%A==0 or x%12!=0)==0:  k=1  if k==0:  print(A)    # Ответ: 36 |
| 16 Вычисление рекуррентных выражений  *Для решения этой задачи лучше написать программу (в комментариях приведено условие из Демо-2021)*  #F(n) = 1 при n = 1;  #F(n) = n + F(n − 1), если n – чётно,  #F(n) = 2 × F(n − 2), если n > 1 и при этом n – нечётно.  #F(26)-?  def f(n):  if n==1:  return 1  elif n % 2 == 0:  return n+f(n-1)  else:  return 2\*f(n-2)  print(f(26)) |
| 17 Умение создавать собственные программы (20–40 строк) для обработки целочисленной информации  *Для решения этой задачи лучше написать программу*  st=100001  fn=900009  m,k=0,0  for x in range(st,fn+1):  if x%11==0 and x%55!=0 and (x%7+x%10==10):  k+=1  m=max(m,x)  print(m,k) |
| 18 Умение обрабатывать вещественные выражения в электронных таблицах  *Задачу лучше сделать в ЭТ* |
| 19-21 Умение анализировать алгоритм логической игры и найти выигрышную стратегию  Данная задача может быть решена в электронных таблицах. Метод, предложенный Артёмом Имаевым, можно посмотреть по ссылке: [https://youtu.be/iQ8OFqML2b0](https://vk.com/away.php?to=https%3A%2F%2Fyoutu.be%2FiQ8OFqML2b0&cc_key=) |
| 22 Умение анализировать алгоритм, содержащий ветвление и цикл  *Для решения этой задачи можно устроить перебор значений (главное определиться с интервалом перебора), аналогично заданию №6*  Например, ниже записана программа, которая вводит натуральное число x, выполняет преобразования, а затем выводит одно число. Укажите наименьшее возможное значение x, при вводе которого программа выведет число 120.   |  |  | | --- | --- | | Код программы | Решение | | x = int(input())  a = 1  while x > 0:  a \*= x % 11  x = x // 11  print(a) | for i in range(1,1000):  x = i  a = 1  while x > 0:  a \*= x % 11  x = x // 11  if a==120:  print(i) | |
| 23 Умение анализировать результат исполнения алгоритма  Эту задачу можно решать различными способами.   1. Можно делать письменно на листке бумаги 2. Можно написать рекурсивный алгоритм 3. Можно написать вспомогательную программу с использованием массива   #У исполнителя есть две команды, которым присвоены номера:  # 1. Прибавить 1  # 2. Умножить на 2  #Сколько существует программ, для которых при исходном числе 1 результатом  #является число 20, и при этом траектория вычислений содержит число 10?  a=[0]\*25  a[1]=1  a[2]=2  for i in range(3,11):  if i % 2 == 0:  a[i]=a[i-1] + a[i // 2]  else:  a[i]=a[i-1]  for i in range(11,20):  a[i]=a[10]  a[20]=a[19] + a[10]  print(a[20])   1. Можно решить в электронных таблицах. На рисунке показано решение для задачи с таким условием: Исполнитель преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:   1. Прибавить 1  2. Умножить на 2  3. Умножить на 3  Сколько существует программ, которые преобразуют исходное число 2 в число 36, и при этом траектория вычислений содержит число 12 и не содержит числа 30? |
| 24 Умение создавать собственные программы (10–20 строк) для обработки символьной информации  *Задачи вида*   |  |  | | --- | --- | | f=open('24.txt')  k=0  for line in f.readlines():  if line.count('A')>line.count('E'):  k+=1  print(k) | f=open('24.txt')  s=f.readline()  k,m=1,0  for i in range(1,len(s)):  if s[i-1]==s[i]:  k+=1  m=max(m,k)  else:  k=1  print(m) | |
| 25 Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации  *Найти количество чисел из заданного отрезка, имеющих ровно два делителя не считая 1 и самого числа*  st=185348  fn=185399  for i in range(st,fn+1):  k=0  for j in range(2,i//2+1):  if i%j==0:  k+=1  if k==1:  d1=j  elif k==2:  d2=j  elif k>2:  break  if k==2:  print(d1,d2) |
| 26 Умение обрабатывать целочисленную информацию с использованием сортировки  *Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов. Известно, какой объём занимает файл каждого пользователя. По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.*  *Входные данные.*  *В первой строке входного файла находятся два числа: S – размер свободного места на диске (натуральное число, не превышающее 10 000) и N – количество пользователей (натуральное число, не превышающее 1000). В следующих N строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.*  *Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.*  f=open('26.txt')  s,n=map(int,f.readline().split())  a=[]  for i in range(n):  a.append(int(f.readline()))  a.sort()  sum,i=0,0  while sum<=s:  sum+=a[i]  i+=1  i-=1  sum-=a[i]+a[i-1]  m=i  while sum+a[m]<=s:  m+=1  print(i,a[m-1])  Эта задача, как правило, также легко решается в электронных таблицах |
| 27 Умение создавать собственные программы (20–40 строк) для анализа числовых последовательностей  *Задача аналогичная вариантам прошлых лет. Для задачи А возможно переборное решение, которое может отработать и на файле к задаче В за более длительное время.*  *На вход программы поступает последовательность из N целых положительных чисел. Рассматриваются все пары различных элементов последовательности. Необходимо определить количество пар чисел, разность которых кратна 13, а произведение чётно.*  *Входные данные.*  *Даны два входных файла (файл A и файл B), каждый из которых содержит в первой строке количество чисел N (2 ≤ N ≤ 12000). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10 000.*  *Программа должна вывести в первой строке одно число: количество пар чисел, разность которых кратна 13, а произведение чётно. Если подходящих пар нет, нужно вывести “NO”.*  f=open('27-A.txt')  n=int(f.readline())  k=0  a=[]  for i in range(n):  a.append(int(f.readline()))  for i in range(n-1):  for j in range(i+1,n):  if (a[i]-a[j]) % 13 == 0 and (a[i]\*a[j]) % 2 ==0:  k+=1  print(k)  Эту задачу часто можно решить с помощью электронных таблиц |