

Информатика

Открытый билет 11 класс

1. В ящике лежат красные, белые и черные кубики. Сообщение о том, что достали красный кубик, несет 5 бит информации. Вероятность извлечения черного кубика в 2 раза больше, чем красного. Сколько информации несет сообщение об извлечении черного кубика? Запишите решение.

(5 баллов)

Решение:

Используем в решении формулу Хартли: $I = \log_2 N = \log_2 \frac{1}{p}$. Введем следующие

обозначения:

p_k – вероятность достать красный шар;

p_q – вероятность достать черный шар;

I_k – количество информации в сообщении о том, что достали красный шар;

I_q – количество информации в сообщении о том, что достали черный шар.

Тогда по условию задачи:

$$5 = \log_2 \frac{1}{p_k}$$

$$I_q = \log_2 \frac{1}{p_q} = \log_2 \frac{1}{2p_k} = \log_2 \frac{1}{2} + \log_2 \frac{1}{p_k} = -1 + 5 = 4.$$

Ответ: сообщение об извлечении черного кубика несет 4 бита информации.

2. Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит ровно 2 единицы.

1) 7

2) 11

3) 12

4) 15

Запишите решение.

(5 баллов)

Решение:

Выполним перевод чисел в двоичную систему счисления

$$\begin{array}{r} 7 \\ \hline 6 \end{array} \begin{array}{r} 2 \\ \hline 3 \end{array} \begin{array}{r} 2 \\ \hline 1 \end{array} \begin{array}{r} 2 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 11 \\ \hline 10 \end{array} \begin{array}{r} 2 \\ \hline 5 \end{array} \begin{array}{r} 2 \\ \hline 4 \end{array} \begin{array}{r} 2 \\ \hline 2 \end{array} \begin{array}{r} 2 \\ \hline 2 \end{array} \begin{array}{r} 2 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 12 \\ \hline 12 \end{array} \begin{array}{r} 2 \\ \hline 6 \end{array} \begin{array}{r} 2 \\ \hline 6 \end{array} \begin{array}{r} 2 \\ \hline 3 \end{array} \begin{array}{r} 2 \\ \hline 2 \end{array} \begin{array}{r} 2 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 15 \\ \hline 14 \end{array} \begin{array}{r} 2 \\ \hline 7 \end{array} \begin{array}{r} 2 \\ \hline 6 \end{array} \begin{array}{r} 2 \\ \hline 3 \end{array} \begin{array}{r} 2 \\ \hline 2 \end{array} \begin{array}{r} 2 \\ \hline 1 \end{array}$$

$$7_{10} = 111_2$$

$$11_{10} = 1011_2$$

$$12_{10} = 1100_2$$

$$15_{10} = 1111_2$$

Ответ: ровно две единицы содержит двоичная запись числа 12.

3. Запись числа 30_{10} в системе счисления с основанием N оканчивается на 0 и содержит 4 цифры. Чему равно основание этой системы счисления N . Запишите решение.

Решение:

Начнем перебирать основания систем счисления с $N = 2$.

$N = 2$: Двоичная запись числа 30_{10} содержит пять цифр, что не соответствует условию задачи.

$N = 3$: Так как число 30_{10} делится на 3 без остатка, то запись числа в системе счисления с основанием 3 будет оканчиваться на 0. Найдем количество цифр искомого числа: $3^3 < 30 < 3^4$, тогда в троичной записи числа 30_{10} будет четыре цифры. Таким образом, все условия задачи соблюдены, основание системы счисления $N = 3$.

Ответ: основание системы счисления $N = 3$.

4. На числовой прямой даны три отрезка: $P = [5,25]$, $Q = [5,15]$ и $R = [10,20]$. Выберите такой интервал A , что формулы

$$(x \notin A) \rightarrow (x \notin P) \quad \text{и} \quad (x \notin Q) \rightarrow (x \in R)$$

тождественно различны, то есть принимают разные значения при любом значении переменной x (за исключением, возможно, конечного числа точек).

$$1) (5, 12) \quad 2) (10, 18) \quad 3) (18, 25) \quad 4) (20, 35)$$

Здесь:

\rightarrow — операция следования, \notin/\in — не принадлежит/принадлежит

Запишите решение.

(10 баллов)

Решение:

Введем следующие обозначения:

$$A \text{ — } (x \in A), \quad \neg A \text{ — } (x \notin A), \quad P \text{ — } (x \in P), \quad \neg P \text{ — } (x \notin P), \\ Q \text{ — } (x \in Q), \quad \neg Q \text{ — } (x \notin Q), \quad R \text{ — } (x \in R), \quad \neg R \text{ — } (x \notin R).$$

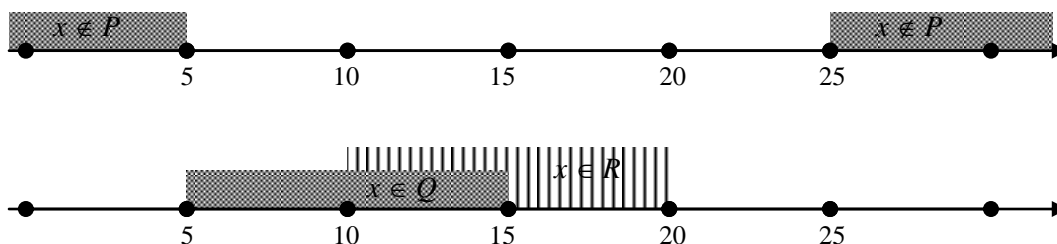
Перепишем формулы с учетом введенных обозначений:

$$\neg A \rightarrow \neg P \quad \text{и} \quad \neg Q \rightarrow R$$

Начнем решение с преобразования формул, заменив операцию следования логическим ИЛИ (\vee) $X \rightarrow Y = \neg X \vee Y$:

$$\neg \neg A \vee \neg P \quad \text{и} \quad \neg \neg Q \vee R; \\ A \vee \neg P \quad \text{и} \quad Q \vee R$$

Отметим интервалы, определяемые формулами, на числовой прямой:



Необходимо выбрать интервал A таким образом, чтобы заштрихованные интервалы на первом и втором рисунке не перекрывались. Рассмотрим имеющиеся варианты:

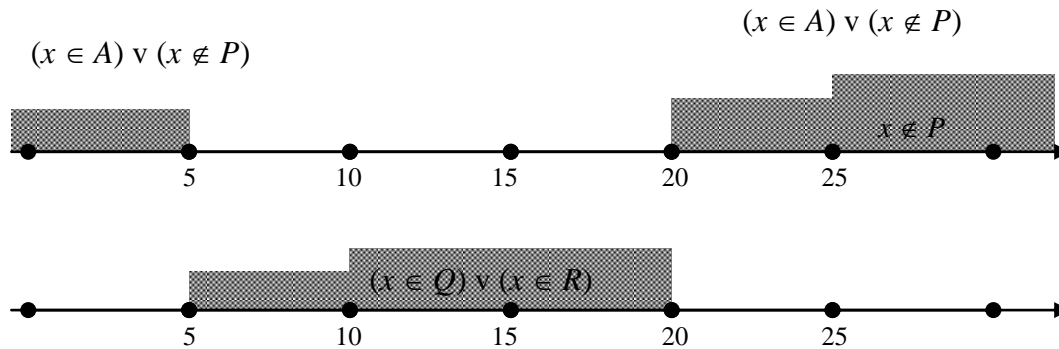
- 1) $(5, 12)$: не подходит по условию задачи, поскольку в этом случае формулы будут

принимать одинаковые значения на интервале (5,12).

2) (10, 18) : не подходит по условию задачи, поскольку в этом случае формулы будут принимать одинаковые значения на интервале (10,18).

3) (18, 25) : не подходит по условию задачи, поскольку в этом случае формулы будут принимать одинаковые значения на интервале (18,20).

4) (20, 35) : правильный ответ, при этом значении интервала A формулы будут тождественно различны, и это хорошо видно на следующем рисунке:



Ответ: $A = (20, 35)$

5. Определите значение переменной c после выполнения следующего фрагмента программы

Бейсик $a = -3$ $a = -a - a$ $b = a + 2$ $c = -b + a * b$	Паскаль $a := -3;$ $a := -a - a;$ $b := a + 2;$ $c := -b + a * b;$	СИ $a = -3;$ $a = -a - a;$ $b = a + 2;$ $c = -b + a * b;$
--	---	--

1) 12

2) 32

3) 40

4) 48

(5 баллов)

Решение:

Выполним указанный фрагмент программы, записывая в таблицу результаты

Шаг	1	2	3	4
Переменная				
a	-3	$-(-3)-(-3) = 6$	6	6
b	-	-	$6+2 = 8$	8
c	-	-	-	$-8+6*8 = 40$

Ответ: $c = 40$.

В заданиях 6-9 требуется написать программу на алгоритмическом языке (СИ, Паскаль, Бейсик, Python, алгоритмический).

6. Дано целое число $N (> 2)$ и набор из N чисел — значений некоторой величины, полученных в N опытах. Найти среднее значение этой величины. При вычислении среднего значения не учитывать минимальное и максимальное из имеющихся в наборе значений.

Для решения этого задания следует использовать алгоритм, позволяющий получить требуемый результат после **однократного** просмотра набора исходных данных.

(15 баллов)

Решение:

Для решения этой задачи реализуем следующий алгоритм:

Опишем целочисленную переменную N (количество опытов) и запросим ее значение с клавиатуры.

Опишем переменную a , которую в дальнейшем будем использовать для ввода значений величины, переменные max и min для хранения текущих значений максимальной и минимальной величин, переменную S , в которой будем накапливать значение суммы.

Инициализируем начальное значение переменной $S = 0$.

Организуем ввод первого значения величины и сохраним считанное значение в переменных max и min .

Увеличим значение переменной S на считанное значение a .

Ввод оставшихся значений величины организуем в цикле.

Каждое введенное значение будет сравниваться с текущими значениями переменных min и max . Если текущее значение величины станет больше значения переменной max , текущее значение max будет изменено, если текущее значение величины будет меньше переменной min , ее значение так же будет изменено.

Каждое введенное значение будет добавляться к текущему значению переменной S .

Цикл будет продолжаться $N-1$ раз.

После цикла значение переменной S уменьшим на значения переменных min и max и разделим на $N-2$ для получения среднего значения.

Решение на языке Си представлено ниже:

```
1 #include <conio.h>
2 #include <stdio.h>
3 int main(){
4     system("chcp 1251");
5     int N,a,min,max,i;
6     float S = 0;
7     printf("Введите количество опытов: ");
8     scanf("%d",&N);
9     printf("Введите значения величин: ");
10    scanf("%d",&a);
11    max = a; min = a;
12    S = S + a;
13    for(i=1;i<=N-1;i++){
14        scanf("%d",&a);
15        if (max < a) max = a;
16        if (min > a) min = a;
17        S = S + a;
18    }
19    S = S - min - max;
20    S = S/(N-2);
21    printf("Среднее значение величины без учета максимального и минимального значения %f\n: ",S);
22 }
```

7. Дан одномерный массив размера N (N — четное число). Элементы массива — целые числа. Найти номера двух ближайших элементов из этого массива (то есть элементов с наименьшим модулем разности) и вывести эти номера в порядке возрастания. Вспомогательный массив не использовать.

(15 баллов)

Решение:

Для решения этой задачи реализуем следующий алгоритм. Опишем целочисленную переменную N для хранения количества элементов массива. Опишем статический массив X из 100 элементов. Организуем ввод значения размерности массива с клавиатуры. Проверим, что введенное с клавиатуры значение не превышает размера статического массива. Для организации ввода элементов массива с клавиатуры используем цикл, который будет выполняться N раз.

Для нахождения элементов массива, модуль разности которых минимален необходимо

найти модули разности всех возможных пар элементов и выбрать из них минимальный. Для сохранения номеров указанных элементов опишем в программе переменные $i1$ и $i2$. Будем считать ближайшими элементами нулевой и первый элементы массива, то есть определим $i1 = 0$, $i2 = 1$, в переменной min сохраним модуль разности элементов $X[0]$ и $X[1]$. Для перебора всех возможных пар элементов организуем два вложенных цикла. Внешний цикл будет перебирать элементы от нулевого до $N-1$ (от первого до последнего) изменяя счетчик i , внутренний цикл будет перебирать элементы от $i+1$ -го (чтобы исключить обращение к ранее просмотренным парам элементов) до последнего элемента, изменяя счетчик j . Во внутреннем цикле организуем вычисления модуля разности элементов $X[i]$ и $X[j]$ и сравнение найденного модуля с текущим значением переменной min . Если найденный модуль разности меньше текущего минимального значения, переопределим значения переменных min , $i1$ и $i2$.

После окончания циклов выведем значения переменных $i1$ и $i2$. Дополнительной проверки для сравнения найденных индексов можно не предусматривать, поскольку из-за описанной выше комбинации циклов в переменная $i1$ всегда будет меньше переменной $i2$. Реализация алгоритма на языке Си приведена ниже. \

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int X[100], N, b, i, j, i1=0, i2=1;
    system("chcp 1251");
    printf("Введите размерность массива: ");
    scanf("%d", &N);
    if (N > 100) {printf("Ошибка...");
                return 0;
    }
    printf("Вводите элементы массива: ");
    for(i=0; i<N; i++)
        scanf("%d", &X[i]);
    int min = abs(X[0]-X[1]);
    printf("\n");
    for(i=0; i<N; i++){
        for(j=i+1; j<N; j++){
            b = abs(X[i]-X[j]);
            if (b < min){ min = b; i1=i; i2=j;
            }
        }
    }
    printf("Ближайшие элементы - %d и %d \n", i1, i2);
    return 0;
}
```

8. Задан двумерный массив $M(7,5)$. Элементы массива – целые числа. Определить номер строки, в которой находится максимальное количество чисел из интервала от -2 до 2.

(15 баллов)

Решение:

Для решения этой задачи реализуем следующий алгоритм. Опишем в программе двумерный массив $M[7][5]$. Опишем переменную max для сохранения максимального количества чисел из интервала $[-2,2]$. В начале работы присвоим переменной max значение 0. Опишем переменную k для сохранения количества указанных элементов в текущей строке матрицы. Опишем переменную $imax$ для сохранения номера строки с максимальным количеством чисел из указанного интервала. В начале работы присвоим переменной $imax$ значение -1. Если после

просмотра массива значение *imax* останется отрицательным, можно будет сделать вывод о том, что в массиве отсутствуют элементы из указанного интервала.

Организуем для ввода элементов массива два вложенных цикла. Внешний цикл изменяет счетчик *i*, тем самым перебирая строки двумерного массива. Перед началом внутреннего цикла будем обнулять значение переменной *k* для накопления количества элементов из интервала $[-2,2]$ в каждой строке матрицы. Внутренний цикл будет изменять счетчик *j* и перебирать элементы строки с номером *i*. Во внутреннем цикле выполним ввод элемента двумерного массива с индексами *i* и *j*. Если введенный элемент принадлежит интервалу $[-2,2]$, то будем наращивать значение переменной *k* на единицу.

После выполнения внутреннего цикла сравним найденное значение переменной *k* со значением текущего максимума, если найденное значение больше, то переопределим значения переменных *max* и *imax*.

После выполнения внешнего цикла выведем на экран найденное значение переменной *imax*.

Ниже представлен код программы на языке Си.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int M[7][5], k, imax=0, i, j, max=0;
    system("chcp 1251");
    printf("Вводите элементы массива: ");
    for(i=0; i<7; i++){
        k=0;
        for(j=0; j<5; j++){
            scanf("%d", &M[i][j]);
            if (M[i][j]>=-2&&M[i][j]<=2) k++;
        }
        if (k>max){ max=k; imax = i; }
    }
    printf("Номер строки - %d, количество элементов из интервала [-2,2] - %d \n", imax, max);
    return 0;
}
```

9. Специальная камера, установленная на перекрестке, фиксирует количество проезжающих автомобилей, и каждую минуту по каналу связи передает неотрицательное целое число — количество автомобилей, проехавших перекресток за эту минуту. Известно, что за минуту перекресток может проехать не более 100 автомобилей. Необходимо найти в заданной серии показаний максимальное количество автомобилей, проехавших перекресток в течение пяти подряд идущих минут. Максимальное количество показаний, которое может передать камера, не превышает 1440.

Напишите на любом языке программирования программу для решения поставленной задачи. Для получения максимального результата программа должна быть эффективна по времени и по используемой памяти.

Входные данные представлены следующим образом. В первой строке задается число *N* — общее количество переданных показаний. Гарантируется, что $N > 5$. В каждой из следующих *N* строк задается одно положительное целое число — очередное показание камеры.

Пример входных данных:

```
8
5
12
27
10
4
50
7
16
```

Программа выводит только одно число — наибольшее количество автомобилей, проехавших перекресток за пять подряд идущих минут.

Решение:

Для решения этой задачи реализуем следующий алгоритм.

Опишем в программе переменную N для хранения количества входных данных. Для оптимизации программы по памяти будем хранить только серию из пяти текущих значений, для этого опишем в программе массив для хранения только пяти переменных – $X[5]$. Организуем цикл для ввода пяти первых значений в массив X . За максимальное количество машин, примем количество машин, проехавших перекресток за первые пять минут, для этого опишем переменную max и сохраним в ней сумму элементов массива. Это же значение сохраним в переменной sum . В дальнейшем переменную sum , будем использовать для хранения количества машин, проехавших перекресток в течение текущих пяти минут.

Для организации ввода оставшихся данных используем цикл, изменяющий счетчик i , который будет повторяться $N-5$ раз. Опишем переменную a , в которую будем вводить значение количества машин, проехавших перекресток в i -тую минуту. В цикле будем выполнять следующие действия:

- ввод количества машин;
- уменьшение переменной sum на значение $X[0]$ и увеличение на значение a (таким образом организовано вычисление количества машин, проехавших перекресток за текущие 5 минут);
- сдвиг элементов в массиве X на одну позицию влево и заполнение последнего элемента массива считанным значением a ;
- сравнение значения sum с текущим значением максимальной суммы и изменение значения переменной max , если значение $sum > max$.

После выполнения суммы организуем вывод переменной max .

Ниже представлен код программы на языке Си.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {
    int X[5], max, sum = 0, i, a, N, j;
    system("chcp 1251");
    printf("Введите количество показаний: ");
    scanf("%d", &N);
    printf("Вводите показания: ");
    for(i=0; i<5; i++){
        scanf("%d", &X[i]);
        sum=sum+X[i];
    }
    max = sum;
    for(i=0; i<N-5; i++){
        scanf("%d", &a);
        sum = sum-X[0]+a;
        if (sum>max) max = sum;
        // сдвиг элементов массива данных влево на одну позицию
        for (j=0; j<4; j++)
            X[j]=X[j+1];
        X[j] = a;
    }
    printf("Максимальное количество машин, проехавших перекресток за 5 минут - %d \n", max);
    return 0;
}
```