

Решение открытого билета по информатике

1. Решение:

Количество символов, содержащихся в предложении – 91. Так как по условию задачи каждый символ кодируется 8 битами, информационный объем предложения равен $91 * 8 = 728$ бит. 1 байт равен 8-ми битам, следовательно, верный ответ 4) – 91 байт.

Ответ: 4 (91 байт)

2. Решение:

Переведем целую часть числа из десятичной системы счисления в восьмеричную:

$$\begin{array}{r|l} 14 & 8 \\ \hline 8 & 1 \\ \hline & 6 \end{array}$$

$$14_{10} = 16_8$$

Переведем дробную часть числа из десятичной системы счисления в восьмеричную:

$$\begin{array}{r} \times 0, 18 \\ \quad 8 \\ \hline \times 1, 44 \\ \quad 8 \\ \hline 3, 52 \end{array}$$

$$0,18_{10} = 0,13_8$$

Ответ: 16,13₈

3. Решение:

$$\begin{array}{r} \cdot 10 \\ \underline{1011011} \\ 10101 \\ \underline{ 10101} \\ 1000110 \end{array}$$

Ответ 1000110₂

4. Решение:

Определим порядок действий:

$$A \& (B \vee \bar{C}) \& C$$

Выполним операции в указанном порядке и результаты запишем в таблицу:

A	B	C	\bar{C}	$B \vee \bar{C}$	$A \& (B \vee \bar{C})$	$A \& (B \vee \bar{C}) \& C$
0	0	0	1	1	0	0

0	0	1	0	0	0	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	1

5. Решение:

Проанализируем логическое выражение:

Выражение истинно только в случае истинности следующих выражений:

$$1) \neg(A = B)$$

$$2) (A > B) \rightarrow (B > C)$$

$$3) (B > A) \rightarrow (C > B)$$

По определению импликации – импликация ложна, только если из истинного выражения следует ложное выражение.

Подставляем имеющиеся варианты:

1) В = 42

- Выражение 1) – истинно $\neg(45 = 42)$
- Выражение 2) - ложно $(45 > 42) \rightarrow (42 > 43)$
- Выражение 3) можно не проверять.

2) В = 43

- Выражение 1) – истинно $\neg(45 = 43)$
- Выражение 2) - ложно $(45 > 43) \rightarrow (43 > 43)$
- Выражение 3) можно не проверять.

3) В = 44

- Выражение 1) – истинно $\neg(45 = 44)$
- Выражение 2) - истинно $(45 > 44) \rightarrow (44 > 43)$
- Выражение 3) - истинно $(44 > 45) \rightarrow (43 > 44)$

4) В = 45

- Выражение 1) – ложно $\neg(45 = 45)$
- Выражения 2) и 3) можно не проверять

5) В = 46

- Выражение 1) – истинно $\neg(45 = 46)$
- Выражение 2) - истинно $(45 > 46) \rightarrow (46 > 43)$
- Выражение 3) - ложно $(46 > 45) \rightarrow (43 > 46)$

Ответ: В = 44.

6. Решение:

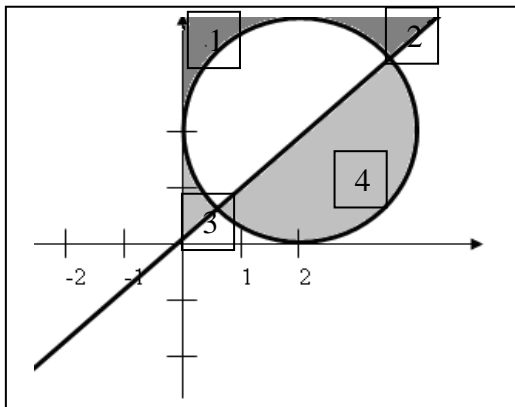
Выполним указанный фрагмент программы, записывая в таблицу результаты

Шаг	0	1	2	3	4
Переменная					
S	0	1	17	66	166
I	1	4	7	10	13
I<=10	истина	истина	истина	истина	ложь

Ответ: S = 166

7. Решение:

Обозначим области на рисунке следующим образом:



Анализ рисунка показывает, что для областей 1,2,3 можно построить одно сложное условие:

$X \geq 0$ и (точка должна находиться правее оси OY)

$Y \leq 4$ и (точка должна находиться ниже прямой $Y = 4$)

$Y \geq X$ и (точка должна находиться выше прямой $Y = X$)

$(X-2)^2 + (Y-2)^2 > 4$ (точка должна лежать за окружностью).

Для области 4 условие выглядит следующим образом:

$Y \leq X$ и (точка должна находиться ниже прямой $Y = X$)

$(X-2)^2 + (Y-2)^2 \leq 4$ (точка должна лежать в окружности)

Тогда условие вхождения точки в заданную область можно описать следующим образом:

$$X \geq 0 \text{ И } Y \leq 4 \text{ И } Y \geq X \text{ И } (X-2)^2 + (Y-2)^2 > 4$$

ИЛИ

$$Y \leq X \text{ И } (X-2)^2 + (Y-2)^2 \leq 4$$

Запишем решение задачи на языке **C++**:

```
int main(int argc, char *argv[])
{
    printf("Введите координаты точки: ");
    float x,y;
    scanf("%f%f",&x,&y); // ввод координат точки с клавиатуры
    if (x>=0&&y<=4&&y>=x&&((x-2)*(x-2)+(y-2)*(y-2))>=4 //
        y<=x&&((x-2)*(x-2)+(y-2)*(y-2))<=4)
        printf("Точка принадлежит выделенной области");
    else printf("Точка не принадлежит выделенной области");
    return 0;
}
```

8. Решение:

Для решения задачи будем использовать свойство перемножения чисел: если перемножаются разные по знаку числа, то результат будет отрицательным, в противном случае результат будет положительный. Если при перемножении получен 0, то хотя бы один сомножитель равен 0.

Алгоритм решения можно описать следующим образом:

Присвоим переменной *t* значение **True**. Организуем просмотр элементов массива от первого до предпоследнего. Если произведение $X[i]*X[i+1] \geq 0$ то переменной *t* присвоить значение **False** и досрочно закончить выполнение цикла.

Программа на языке **Cu**:

```
int main(int argc, char *argv[])
{
    float a[20]; // описание массива из 20 элементов
    int i;
    bool t = true;
    printf("Вводите элементы массива: ");
    for(i=0;i<20;i++)
        scanf("%f", &a[i]); // чтение элементов массива
    for(i=0;i<19;i++) // проверка значения произведения рядом стоящих элементов
        if(a[i]*a[i+1]>=0) {
            t = false;
            break; // досрочный выход из цикла
        }
    if (t) printf("t - true\n"); else printf("t-false\n");
    return 0;
}
```

9. Решение:

Для решения задачи будем использовать функцию *strtok* (для выделения слов) и функцию *strlen* (для определения размера слова) языка программирования **Cu**.

```
int main(int argc, char *argv[])
{
    system("chcp 1251");
    printf("Вводите строку: ");
    char str[256]; // описание строки
    gets(str); // ввод строки с клавиатуры
    int c = 0; // обнуляем счетчик слов
    char *word = strtok(str, " "); // выделяем первое слово
    while (word!=NULL){ // если слово найдено
        // сравниваем первый и последний символы слова
        if (word[0]==word[strlen(word)-1]) c++; //если символы равны – увеличиваем
        счетчик
        word = strtok(NULL, " "); // выделяем следующее слово
    }
}
```

```

    printf("Количество слов, первый и последний символы которых совпадают -
    %d",c);
    return 0;
}

```

10. Решение:

Если исходное число меньше 10, исходное число – цифровой корень, выведем его на экран и закончим работу алгоритма, в противном случае будем выполнять следующие действия:

- Преобразуем исходное число в строку.
- Найдем количество символов полученной строки.
- Преобразуем все символы строки в числа и найдем их сумму.
- Найденную сумму примем за исходное число.
- Повторим работу алгоритма с начала.

Программа на языке **Cu**:

```

int main(int argc, char *argv[])
{
    int N,i;
    printf("Введите число: ");
    scanf("%d",&N); // чтение числа
    char str[10]; // строка для хранения числа
    char r[2]; // вспомогательная строка для выделения цифр числа
    // пока число больше девяти
    while(N>9){
        itoa(N,str,10); // преобразовать число в строку
        N = 0; //обнулить сумму
        // просчитать сумму цифр числа
        for(i = 0;i<strlen(str);i++)
            { r[0] = str[i];
              r[1] = '\0';
              N+=atoi(r);
            }
    }
    printf("Цифровой корень - %d\n",N);
    return 0;
}

```