**Примеры ошибок в выполнении олимпиадных заданий 1 тура**

**Задание 1.**

Разработать программу для вывода на экран всех целых чисел в диапазоне от 1 до 1000, квадраты которых равны кубу **другого числа** в том же диапазоне.

Лишний результат:

1 \* 1 = 1

1 \* 1 \* 1 = 1

1^2=1 и 1^3=1

**Задание 2**

Разработать программу, которая вычисляет значение логарифма вводимого числа А по заранее заданному основанию В. На вход программы подаются значение основания В логарифмической функции и логарифмируемое число А. Программа выводит значение С - логарифм числа А по основанию В.

Пример 1

В программе вычисления логарифма числа необходимо проверять условие:

 if (b>0) and (b<>1) and (a>0) then …

Примеры ошибок:

Введите число, логарифм которого нужно получить, и основание, по которому будет вычислен логарифм

Число:-257

Основание:2

» Ошибка: неверная вещественная операция (z2u31.pas, строка 12)

Пример 2

Необходимо учитывать, что в программной реализации функция логарифма не во всех средах программирования умеет преобразовывать выражения:

![\[{\log _{{a^n}}}{x^m} = \frac{m}{n}{\log _a}x\]]()

Введите число, логарифм которого нужно получить, и основание, по которому будет вычислен логарифм.

Число:256

Основание: 0.5

» Ошибка ввода. Программа завершена (z2u31.pas, строка 10)

Желательно устанавливать условия и предупреждения на допустимые данные (аргументы функции).

Пример 3

Иногда, программа верно получает часть результатов, но в общем виде алгоритмически решение задачи не завершено.

while A>1 do

begin

x:=(A/B);

A:=x;

C:=C+1{считает степень числа}

end;

Введите основание В

2

Введите число А

17

Ответ 5

**Задание 3**

Разработать программу, которая определяет количество информации по формуле Р.Хартли. На вход программы подается N - количество равнозначных возможных состояний **двоичного слова из Н разрядов**. Программа выводит количество информации Н, которое вмещает один символ N-элементного алфавита.

N = 256

H = 8

Пример исполнения программы, не соответствующей условиям задания:

N = 257

H = 9

**Задание 4**

Разработать игровую программу, которая предназначена для угадывания кода символа. Программой на экран случайным образом выводится символ из кодовой таблицы символов. Игрок вводит код символа, программа реагирует выводом одного из трех вариантов: Больше. Меньше. Поздравляю, угадали! Игроку дается 3 попытки, после чего на экран выводится код символа. После каждого угадывания выводится количество правильных и неправильных угадываний.

Не считается ошибкой, что больше-меньше *,* код загадываемого символа или код отвечающего.

В задании используется понимание кода из кодовой таблицы (например ASCII), как это принято в информатике. Если Вы используете свое кодирование, объясните это в условии игры.

Игра основывается на знании кодов. Угадать номер буквы в приведенном примере за 3 попытки маловероятно.

Пример:

хотите поиграть? (y/n): y

У вас 3 попытки!

Ваша строка - abcdefghijklmnopqrstuvwxyz

ваша цель - угадать номер загаданного символа

Введите номер символа:15

Номер загаданного числа меньше!

Введите номер символа:7

Номер загаданного числа больше!

Вы не угадали, индекс символа был 8, а символ был " h ".

По логике задания программой выводится символ из кодовой таблицы символов, а игрок угадывает его код (номер в стандартной кодовой таблице).

Необходим так же вывод количества правильных и неправильных угадываний (правильных и всех угадываний).

**Задание 5**

Разработать игровую программу для тренировки памяти. В программе хранится 25 произвольных слов из учебника информатики. За один тайм игры на экран последовательно, по нажатию на клавишу Enter, выводится каждое из пяти слов. После вывода следующего слова, предыдущее удаляется с экрана. После просмотра последнего из 5 слов, игроку предлагается последовательно ввести эти слова: Напечатайте 1 слово, Напечатайте 2 слово... Программа реагирует на ввод каждого слова выводом одного из двух вариантов: Не угадали. Поздравляю, угадали! После каждого тайма, программа выводит количество угаданных слов.

Программа, использующая все 5 таймов игры, оценивается на 1 балл больше.

Соответственно, если, программа использует только 5 слов, а не все 25, и. требует для каждого следующего тайма повторного запуска, оценивается на 1 балл меньше. Если повторный запуск использует случайный выбор слов, при отсутствии других замечаний, можно получить максимальный балл.

Общее замечание для 2 и 3 туров:

Среди конкурсных работ разных участников, иногда, встречаются абсолютно одинаковые программы. Даже если руководитель один, решения участников должны быть индивидуальны, содержать хотя бы частично разные имена и значения переменных. В первом туре это не использовалось для снижения количества баллов за выполненное задание. В следующих турах, максимальный балл за абсолютно одинаковые программы получит только первый по времени отправки участник олимпиады.