Задания на курсовую работу по предмету «Введение в ИТ»

Задание 1. Написать программу расстановки кораблей в игре «Морской бой» с полем 10x10 клеток и 4-мя типами кораблей: однопалубные (4 шт.), двухпалубные (3 шт.), трехпалубные (2 шт.), четырехпалубный (1 шт.). Корабли на игровом поле расставляются случайным образом и не должны соприкасаться друг с другом. Реализовать возможность сохранения игрового поля и его загрузки из файла.

Задание 2. Написать игру «Сапер», с полем 10x10 клеток. Вначале игры мины (10 штук) случайным образом расставляются на поле. Затем, пользователь вводит координаты клетки и она открывается. В открывшейся клетке появляется число (количество мин вокруг клетки) или мина. Если игрок открывает все клетки без мин, то он выигрывает, иначе – проигрывает.

Задание 3. Дан текст HTML-страницы (скачать по ссылке http://tk.ulstu.ru/lib/info/kurs_11.txt). Необходимо выделить текст внутри всех тегов h3: (т.е. текст заключенный между `<h3 ...>` и `</h3>`), вывести его на экран и сохранить в файл (каждое найденное содержимое с новой строки).

Реализовать без использования регулярных выражений (без модуля re).

Задание 4. Дан текст HTML-страницы (скачать по ссылке http://tk.ulstu.ru/lib/info/kurs_11.txt). Необходимо удалить все теги script (т.е. текст, начинающийся со `<script ...>` и заканчивающийся `</script>`), вывести результат обработки на экран и сохранить в файл.

Реализовать без использования регулярных выражений (без модуля re).

Задание 5. Написать игру «крестики-нолики» с размером поля 3x3 клеток (человек против компьютера). Компьютер ставит нолик (или крестик) в свободную позицию случайным образом.

Задание 6. Написать программу анализа и вычисления арифметических действий во введенной строке. Например, можно ввести строку:

$10+(5*2) - 3$

Возможные арифметические действия: сложение, вычитание, умножение, деление. Все числа – целые. Также возможно наличие круглых скобок. Результат вычислений вывести на экран.

Реализовать без использования функции eval и ей подобных.

Задание 7. Имеется список из слов:

['кара', 'мура', 'каре', 'каюр', 'сток', 'тара', 'кафр', 'муха', 'стон', 'срок', 'крюк', 'тура', 'слон', 'урюк', 'кафе', 'урок', 'каюк']

Необходимо взять первое слово «муха» и, выбирая последовательно слова из списка w, отличающиеся на одну букву от предыдущего слова, дойти до слова «слон». Напишите программу, которая находит минимальную последовательность из слов, превращающих муху в слона.

Задание 8. Имеется текстовый файл (скачать по ссылке <http://tk.ulstu.ru/lib/info/cities.txt>), в котором записаны названия городов (каждый с новой строки). Необходимо прочитать содержимое этого файла, сформировать список из названий городов, перемешать его (функция shuffle модуля random) и взять первые 20 названий. На их основе найти самую длинную последовательность из городов, названия которых начинаются на последнюю букву предыдущего города. Если последняя буква: ь, ы или ы, то брать следующую с конца букву.

Задание 9. Имеется словарь с наименованиями предметов и их весом (в граммах):

```
things = {'карандаш': 20, 'зеркальце': 100, 'зонт': 500, 'рубкашка': 300, 'брюки': 1000, 'бумага': 200, 'молоток': 600, 'пила': 400, 'удочка': 1200, 'расческа': 40, 'котелок': 820, 'палатка': 5240, 'брезент': 2130, 'спички': 10}
```

Сергей собирается в поход и готов взвалить на свои хрупкие плечи максимальный вес в N кг (вводится с клавиатуры). Определите наибольшее количество предметов, которые Сергей может положить в рюкзак так, чтобы их суммарный вес не превышал N кг. Все предметы даны в единственном экземпляре. Выведите список предметов из рюкзака.

Задание 10. Необходимо реализовать градиентный алгоритм поиска точки минимума функции f(x). Алгоритм следующий. Выбирается начальная точка, например, $x_0 = 0$ и, затем, на каждой итерации происходит ее коррекция по правилу:

$$x_n = x_{n-1} - \lambda \cdot \frac{\partial f(x)}{\partial x}$$

где x_n - значение аргумента на n-й итерации; x_{n-1} - значение аргумента на предыдущем (n-1)-м шаге; $\lambda = 0,01$ - шаг сходимости градиентного алгоритма; $\frac{\partial f(x)}{\partial x}$ - значение производной функции в точке x_{n-1} .

В программе задается функция f(x) в виде объявления обычной функции. Через декоратор должна вычисляться производная этой функции (численно, по формуле $\frac{\partial f(x)}{\partial x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$ при $\Delta x = 0,001$) в точке x_{n-1} . Затем, значения этой производной должны использоваться в градиентном алгоритме. Число итераций подберите самостоятельно. А в качестве примера можно взять функцию $f(x) = 0,5x^2 - 5x + 2$.

Задание 11. Имеется игровое поле «Крестики-нолики» с некоторой расстановкой, например:

$P = [['x', 0, 'o'], ['x', 0, 0], [0, 0, 0]]$

Определите следующий лучший ход для ноликов. (Заданная расстановка в поле P может быть и другой – это лишь пример.)

Задачу следует решать полным перебором всех возможных вариантов и выбора лучшего из возможных ходов.

Задание 12. Написать программу «Календарь», которая предоставляет пользователю через пункты меню выполнять следующие действия с датами:

- определять возраст человека по введенной дате рождения и текущей дате;
- определять число дней до нового года (31 декабря) по текущей дате;
- определять число дней между двумя разными датами одного года.

Текущая дата вводится один раз в начале программы. А распределение числа дней в месяцах не високосного года, начиная с января, брать следующее: 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31.

Реализовать без использования модуля `datetime` и ему подобных.

Задание 13. Написать программу, которая просит зарегистрироваться пользователя и запрашивает следующую информацию:

- логин (используются: латинские символы, цифры, символ подчеркивания);
- email (обязательны символы @ и точка (.), причем сначала идет @, а потом точка, остальные символы – это латинские буквы, цифры и символ подчеркивания)
- пароль (минимум 10 символов, обязательно наличие хотя бы одной заглавной буквы и символа из списка «?!#\$\$@%»)
- повтор пароля (должен совпадать с предыдущим введенным значением)

Необходимо проверить корректность ввода полей. Если все поля введены верно, то программа выводит сообщение «Регистрация прошла успешно», а иначе – запрашивает данные снова (пока они не будут введены верно).

Реализовать без использования регулярных выражений (без модуля `re`).

Задание 14. Написать программу обработки информации по книгам. Каждая книга должна содержать: автора, название, год издания, цену (вещественное число). В программе должны быть реализованы следующие возможности (через меню):

- добавление новой книги
- удаление книг по автору/году издания
- поиск всех книг по автору/названию

При запуске программы необходимо загружать ранее сформированный список книг из файла, а при завершении программы – сохранение текущего списка в этот же файл.

Задание 15. Написать программу для хранения генеалогического древа. В программе должны быть реализованы следующие возможности (через меню):

- добавление в древо человека и его прямых родственников (маму, папу, бабушек, дедушек, прабабушек, прадедушек и т.д.)
- отображение всего древа на экране
- поиск в древе родственника по его имени
- удаление из древа родственника и всех связанных только с ним предков

При запуске программы необходимо загружать ранее сформированное древо из файла, а при завершении программы – сохранение текущего древа в этот же файл.