



IIC1103 – Introducción a la Programación
1 - 2017

Interrogación 1

Instrucciones generales:

- La interrogación consta de tres preguntas con igual ponderación.
- Solo se reciben consultas sobre el enunciado en los primeros 30 minutos.
- La última página es un recordatorio; no está permitido utilizar material de apoyo adicional.
- Las tres preguntas están pensadas para ser resueltas en forma sencilla con ciclos y condicionales. No es necesario utilizar otros mecanismos más avanzados.
- Si el enunciado de la pregunta no lo indica explícitamente, no es necesario validar los ingresos del usuario. Puedes suponer que no hay errores por ingreso de datos.
- Responde cada pregunta en una hoja separada, marcando tu RUT y el número de pregunta en todas las hojas prestando atención a la dirección del trazo, tal como se muestra en el siguiente ejemplo:

Pregunta 1 (1/3)

Las palabras que contienen **todas** las letras vocales (a, e, i, o, u), **una** sola vez cada una, se conocen como panvocálicas, o en inglés como *supervocalic* (que, además, es una palabra que cumple esta condición). Aún más especiales son las palabras panvocálicas que contienen las vocales en orden (por ejemplo, la palabra en inglés: *abstemious*. Al parecer no hay palabras que cumplan esta condición en el idioma español).

Nota: puedes asumir que las palabras no contendrán acentos.

- a) (15 puntos) Escribe una función `sv(x)`, que retorne `True` si `x` es panvocálica, `False` si no.

Ejemplos:

<code>sv("abuelito")</code>	<code>True</code>	<code>sv("exhaustiva")</code>	<code>False</code>
<code>sv("aeiou")</code>	<code>True</code>	<code>sv("aeiouaeiou")</code>	<code>False</code>
<code>sv("exhaustivo")</code>	<code>True</code>	<code>sv("si")</code>	<code>False</code>

- b) (30 puntos) Escribe una función `sv_orden(x)`, que retorne `True` si `x` es panvocálica y además sus vocales están ordenadas, `False` si no.

Ejemplos:

<code>sv_orden("abuelito")</code>	<code>False</code>	<code>sv_orden("abstemious")</code>	<code>True</code>
<code>sv_orden("aeiou")</code>	<code>True</code>	<code>sv_orden("facetious")</code>	<code>True</code>
<code>sv_orden("hola")</code>	<code>False</code>	<code>sv_orden("exhaustivo")</code>	<code>False</code>

- c) (15 puntos) Una frase puede tener un trozo que sea panvocálico si es que al recortar parte de la frase, ésta cumple la condición de ser panvocálica. Por ejemplo, “Mi abuelo hace unos increíbles pasteles” tiene el trozo “i abuelo” que es panvocálico. Escribe una función `trozo(x)`, que retorne un trozo panvocálico de la frase `x`; sin letras que sobren, por ejemplo para el caso anterior no es válido retornar “Mi abuelo”. Si hay más de uno, puedes retornar cualquiera. Si no hay ninguno, retorna un string vacío.

Ejemplos

- `trozo("El elefante hizo una pirueta")` entrega "ante hizo u"
- `trozo("Fuimos a la lavanderia")` entrega ""

Pregunta 2 (1/3)

Estás implementando un juego en el que cada jugador tiene un conjunto de letras que debe ingresar a un tablero. El tablero tiene una cantidad de puntos en cada celda, por lo que el puntaje de la palabra será la suma de los puntos de todas las celdas usadas por la palabra. Luego, el siguiente jugador tendrá que ingresar su palabra al tablero, pero considerando las palabras que ya han ingresado los jugadores anteriormente. Una palabra se puede colocar horizontalmente (de izquierda a derecha) o verticalmente (de arriba abajo) en el tablero.

Por ejemplo, nota que el siguiente tablero tiene algunas celdas ya ocupadas (con letras y números), y otras desocupadas (que sólo tienen el número). Si el usuario quiere ingresar la palabra “casa”, la mejor ubicación en la que la puede incluir se muestra en el tablero a continuación, sumando 24 puntos.

Tablero:

5	7	s	p	a
5	a	5	5	5
a	5	6	a	s
2	5	6	5	3
1	a	s	9	0
4	5	9	5	2

→

5	7	s	p	a
5	a	5	c	5
a	5	6	a	s
2	5	6	5	3
1	a	s	s	0
4	5	9	a	2

Debes escribir la función `calcular_max_puntos(t,p)`, que dado un tablero `t` y una palabra `p`, entregue el número de puntos máximo que se podría obtener al colocar la palabra en el tablero (o -1 si la palabra no se puede colocar). Un tablero es una lista de listas, que en cada celda contiene una lista con una letra y un número. Una palabra es una lista de caracteres. Puedes asumir que las letras estarán siempre en minúsculas.

Un ejemplo de uso de función es:

```
t = [
    [" ",5], [" ",7], ["s",8], ["p",1], ["a",1],
    [" ",5], ["a",1], [" ",5], [" ",5], [" ",5],
    ["a",2], [" ",5], [" ",6], ["a",5], ["s",3],
    [" ",1], ["a",4], ["s",1], [" ",9], [" ",0],
    ["a",4], ["s",5], ["a",9], [" ",5], [" ",2]]

p = ["c","a","s","a"]

puntos = calcular_max_puntos(t,p) #puntos sera 24
```

Solamente debes escribir esta función, no necesitas escribir ninguna otra parte del juego.

Pregunta 3 (1/3)

Te postulaste como voluntario para colaborar en el próximo DCCenso y has sido seleccionado. Tu misión es elaborar un programa que permita llevar a cabo este proceso. Para ello contarás con un módulo especialmente desarrollado para esto, llamado `dccenso`, cuyas funciones están descritas más abajo.

Específicamente debes:

1. Definir la función `alumno_viejo(n_alumno, limite)` que retorna `True` si y sólo si la entrada `n_alumno` corresponde al número de un alumno que ingresó hace más que `limite` años a la carrera.

El número de alumno es un entero de 8 dígitos, donde los 2 primeros indican el año de ingreso a la carrera.

- `alumno_viejo(11633506, 5)` retorna `True` ya que el alumno ingresó el 2011, lo que supera el límite de 5 años
- `alumno_viejo(17621234, 0)` retorna `False` ya que el alumno ingresó el 2017, lo que no supera el límite de 0 años
- `alumno_viejo(07634321, 10)` retorna `False` ya que el alumno ingresó el 2007, lo que no supera el límite de 10 años

Asume que ningún alumno ingresó antes del 2000.

2. Programar el siguiente flujo:

- Mientras hayan oficinas por censar en el departamento, obtener (desde el módulo `dccenso`) el identificador de una oficina pendiente
- Para una oficina, mientras hayan miembros por censar en ella, obtener (desde el módulo `dccenso`) el número de alumno de un miembro pendiente
- Para un miembro, pedir (al usuario) la información correspondiente y registrarla (usando el módulo `dccenso`). Si su número de alumno indica que lleva más de 5 años en la carrera, se debe pedir el tiempo en años que estima le falta para egresar. En caso contrario, se debe pedir su edad y estado civil

El módulo `dccenso` tiene las siguientes funciones:

- `departamento_listo()`: Retorna `True` si y sólo si cada una de las oficinas del departamento ha sido censada completamente
- `oficina_lista(oficina)`: Retorna `True` si y sólo si cada miembro de `oficina` ha sido censado
- `siguiente_oficina()`: Retorna el *String* identificador de la próxima oficina por censar
- `siguiente_censado(oficina)`: Retorna un entero con el número de alumno del próximo miembro de `oficina` que debes censar
- `registrar_joven(n_alumno, edad, estado_civil)`: Registra la `edad` y el `estado_civil` del miembro correspondiente a `n_alumno`
- `registrar_viejo(n_alumno, tiempo)`: Registra el `tiempo` estimado en años para egresar de la carrera del miembro correspondiente a `n_alumno`

A continuación se muestra una ejecución de ejemplo:

Estamos censando a la oficina P19

Alumno 11633506 por favor acercarse al mesón

Tu número de alumno indica que eres viejo, te corresponde el formulario A

Años para egresar de la carrera: 0

Alumno 15634213 por favor acercarse al mesón

Tu número de alumno indica que eres joven, te corresponde el formulario B

Edad: 19

Estado civil: soltera

Oficina P19 lista

Estamos censando a la oficina P10

Alumno 12635588 por favor acercarse al mesón

Tu número de alumno indica que eres viejo, te corresponde el formulario A

Años para egresar de la carrera: 5

Oficina P10 lista

DCCenso terminado