

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA 1ra. práctica (tipo B) (Primer Semestre 2019)

Indicaciones Generales:

- Duración: 2h 50 min.
- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Si la implementación es significativamente diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Su trabajo deberá ser subido a PAIDEIA en el espacio indicado por los jefes de práctica.

Pregunta 1 (10 puntos)

Un alumno se va de viaje al extranjero y al regresar, tiene la posibilidad de adquirir algunos productos electrónicos para su venta, por lo cual cuenta con una mochila de mano para transportar como máximo 10 artículos. La venta de cada producto le representa una ganancia (g), además cada uno de ellos tiene un peso determinado (p). Debido a temas de aduana, todos los productos elegidos deben ser diferentes y como máximo puede traer un número determinado de productos (M) y por restricciones de la aerolínea el peso no debe exceder un peso definido (W). Considere $M \leq 10$.

Ejemplo de ingreso de datos:

$$M = 6$$

$$W = 25$$

Artículo	Peso (p)	Ganancia (g)
1	8	15
2	2	20
3	10	5
4	10	10
5	5	8
6	5	5

Solución:

Artículos: 1 2 4 5

Ganancia Total: 53

- a) Implemente un programa que devuelva la mejor solución (artículos seleccionados) y la ganancia total máxima, que podrá obtener por los artículos al venderlos. Desde luego debe considerar no exceder el peso máximo W, si en caso hay más de una respuesta que devuelva la máxima ganancia elegir cualquiera de ellas. (8 puntos).
- b) Indique el orden O de la solución desarrollada, la respuesta solo es válida si el programa está completo (2 puntos).

Pregunta 2 (10 puntos)

El juego “Pupiletras” es un pasatiempo que consiste en una cuadrícula u otra forma geométrica rellena con diferentes letras y sin sentido aparente. El juego consiste en descubrir un número determinado de palabras enlazando estas letras de forma horizontal o vertical y en cualquier sentido. Son válidas las palabras tanto de derecha a izquierda como de izquierda a derecha, y tanto de arriba a abajo, como de abajo a arriba.

Se pide elaborar en C:

a. (1.0 pto.) Una función *PopulateGameBoard* que reciba un número entero (*t*) que indica el tamaño de un tablero cuadrado, y retorne mediante el segundo argumento un arreglo bidimensional de caracteres de tamaño *t* x *t* que representa al tablero a llenar con letras en mayúscula de la ‘A’ a la ‘Z’ generadas aleatoriamente de acuerdo con la siguiente sintaxis:

```
void PopulateGameBoard(int t, char tablero [t][t])
```

b. (1.0 pto.) Una función *ShowGameBoard* que reciba el arreglo del tablero e imprima en pantalla su contenido de acuerdo con la siguiente sintaxis:

```
void ShowGameBoard(int t, char tablero [t][t])
```

Si el tamaño de la cuadrícula es 5, el formato de impresión sería:

```
H | G | A | F | Y
D | Q | V | Z | R
A | M | E | N | H
O | G | O | R | F
T | R | W | I | P
```

c. (4.0 ptos.) Una función *SearchWord* que reciba una palabra o cadena de caracteres (*word*), el número de filas *t* del tablero, el *tablero* de tamaño *t* x *t*, y retorne, mediante el cuarto argumento un arreglo de caracteres de 3 elementos, que contiene la posición, en fila y columna, de la primera letra de la palabra en la cuadrícula, así como la dirección de cómo se leería la palabra. En el arreglo de retorno, el primer elemento indica la fila, el segundo elemento indica la columna y el tercer elemento indica el carácter de dirección. Los valores posibles de dirección son ‘U’ (arriba), ‘D’ (abajo), ‘L’ (izquierda) y ‘R’ (derecha). En caso que la palabra no se encontrara en la cuadrícula, deberá retornar -1 para la fila, la columna y la dirección. Debe usar fuerza bruta para realizar la búsqueda solicitada.

La función deberá tener el siguiente prototipo:

```
void SearchWord(char word[], int t, char tablero[t][t], char res[3])
```

d. (2.0 ptos.) Un programa o función *main*, que solicite al usuario el tamaño (*t*) del tablero, muestre el tablero originado por la función *PopulateGameBoard*, pregunte al usuario por la cantidad de palabras a buscar (*n*), se leerán las *n* palabras en mayúscula y luego se mostrará el resultado por cada palabra si está o no en el tablero, así como su posición de la primera letra y su dirección. Debe utilizar las funciones definidas previamente.

El formato de entrada de datos e impresión de salida deberá ser, por ejemplo:

Ingrese el tamaño de la cuadrícula:

5

Bienvenido al juego de sopa de letras

H | G | A | F | Y

D | Q | G | Z | R

A | M | E | N | H

O | G | O | R | F

T | R | W | I | P

Ingrese la cantidad de palabras a buscar:

5

Palabra 1: **TOAD**

Palabra 2: **FROG**

Palabra 3: **TRIP**

Palabra 4: **MEN**

Palabra 5: **AGE**

Palabra 1 - TOAD encontrada en (5,1) - arriba

Palabra 2 - FROG encontrada en (4,5) - izquierda

Palabra 3 - TRIP no encontrada

Palabra 4 - MEN encontrada en (3,2) - derecha

Palabra 5 - AGE encontrada en (1,3) - abajo

e. (2.0 ptos.) Indique el orden *O* de la solución desarrollada, la respuesta solo es válida si el programa está completo.

Notas:

- Para obtener un número aleatorio se puede utilizar la función `rand()` que devuelve un número entero aleatorio entre 0 y el `RAND_MAX` que suele ser 32767. Para usarlo, previamente se debe invocar a `srand(time(NULL))` para que inicialice la semilla. El código ASCII de la 'A' es el 65 y el de la 'Z' el 90.
- Para leer y guardar una cadena de caracteres (solo una palabra) en `cad` de tamaño `t` como máximo, puede hacer lo siguiente:

```
char cad[t];
printf("Introduce una palabra: ");
scanf("%s", cad);
```

- Para copiar el contenido de una cadena `origen` a otra `destino` puede hacer lo siguiente:

```
//Necesita de la cabecera <string.h>

memset(destino, '\0', sizeof(destino)); //Copia el caracter '\0' en todos los caracteres de la cadena destino.

strcpy(destino, origen) //Copia el contenido de origen en destino, destino debe ser lo suficientemente grande.
```

Profesores del curso: Rony Cueva
Johan Baldeón

Pando, 09 de abril del 2019