



Universidad Andrés Bello  
Facultad de Ingeniería  
Ingeniería Civil Informática

## FUNDAMENTOS DE PROGRAMACIÓN SOLEMNE I

Nombre: \_\_\_\_\_ Nota: \_\_\_\_\_

**Prof. Cátedra:** Carlos Contreras Bolton

**Fecha:** 11 de Septiembre de 2014

**Profs. Laboratorio:** Felipe Reyes – Tamara Saéz – Nicolás Venegas – Omar Opazo

### Instrucciones:

- Coloque su nombre a todas las hojas.
- Apague o silencie sus celulares. NO se podrá contestar llamadas ni visualizar el celular. Si realiza alguna de las acciones anteriores obtendrá nota mínima.
- Conteste cada pregunta en hojas separadas.
- Toda copia o intento de copia será calificada con nota mínima.
- Tiene 90 minutos para realizar la prueba.

1	
2	
3	
4	
Suma:	
	+1
Puntaje:	

### Pregunta 1 (15 puntos)

Realice un programa que dado un número se verifique si es un número Palíndromo.

#### Ejemplo:

121 es Palíndromo.

1122 no es Palíndromo.

12721 es Palíndromo.

### Pregunta 2 (15 puntos)

El supermercado Cotito Market ha lanzado una promoción para todos sus clientes que posean la tarjeta Alex Card. La promoción consiste en aplicar un descuento por cada productos que pasan por caja. El primer descuento es de 20 %, y se aplica sobre los primeros  $n$  productos ingresados. Luego, cada descuento es la mitad del anterior, y es aplicado sobre los siguientes  $n$  productos. Por ejemplo, si  $n = 3$  y la compra es de 11 productos, entonces los tres primeros tienen 20 % de descuento, los tres siguientes 10 %, los tres siguientes 5 %, y los dos últimos no tienen descuento.

Realice un programa que pida al usuario ingresar  $n$  y la cantidad de productos, y luego los precios de cada producto. Al final, el programa debe entregar el precio sub-total, el descuento y el precio final después de aplicar el descuento. Si al aplicar el descuento el precio queda con decimales, redondee el valor hacia abajo.

Además, se necesita que el programa le imprima al cajero el vuelto que debe dar y proporcione el desglose en billetes y monedas de la cantidad exacta de pesos. Debe usar los siguientes billetes: 20000, 10000, 5000, 2000 y 1000 pesos y monedas de 500, 100, 50, 10 y 5 pesos y además de 1 peso. A continuación, un ejemplo de la salida del programa:

```

n=3
Cantidad productos: 8
Precio producto 1: 400
Precio producto 2: 800
Precio producto 3: 500
Precio producto 4: 100
Precio producto 5: 400
Precio producto 6: 300
Precio producto 7: 200
Precio producto 8: 500
Sub-Total: 3200
Descuento: 420
Total: 2780
Pago: 20000
Vuelto: 17220
Desglose:
1 billete de 10000 pesos.
1 billete de 5000 pesos.
1 billete de 2000 pesos.
2 monedas de 100 pesos.
2 monedas de 10 pesos.

```

### Pregunta 3 (15 puntos)

Cuando imprimimos un documento, normalmente la primera página es impresa primero, luego la segunda, luego la tercera y así hasta el final. Sin embargo, cuando creamos un folleto plegado, el orden de la impresión debe ser alterado. Un folleto plegado tiene cuatro páginas por hoja, con dos en el frente y dos atrás. Cuando apila todas las hojas en orden, se pliegan y las páginas aparecen en el orden correcto como en un libro regular. Por ejemplo, un folleto de 4 páginas es impreso en una hoja conteniendo en el frente las páginas 4 y 1, y en el reverso las páginas 2 y 3.

Front (Frente)	Back (Atrás)
-----	-----
4   1	2   3
4   1	2   3
-----	-----

Realice un programa que dado un número de páginas para imprimir, generar el orden de impresión. En la Tabla 1 se observa la entrada y salida.

#### Entrada:

La entrada contiene uno o más casos de prueba, seguidos por una línea con un número 0 que indica el final de la entrada. Cada caso de prueba consiste en un entero positivo  $n$  en cada línea, donde  $n$  es el número de páginas a imprimir;  $n$  no es mayor a 100.

#### Salida:

Por cada caso de prueba, despliegue un reporte que indique que páginas son impresas en cada página, como se muestra en el ejemplo. Si el número de páginas no llena completamente las hojas, imprima la palabra *Blank* en lugar de un número. Si el frente o atrás de una hoja es completamente blanca, no genere su salida. La salida debe estar en orden ascendente por hojas, iniciando por el frente hasta el final.

Tabla 1: Ejemplo de entrada y salida.

Entrada	Salida
1 14 4 0	Printing order for 1 pages: Sheet 1, front: Blank, 1 Printing order for 14 pages: Sheet 1, front: Blank, 1 Sheet 1, back : 2, Blank Sheet 2, front: 14, 3 Sheet 2, back : 4, 13 Sheet 3, front: 12, 5 Sheet 3, back : 6, 11 Sheet 4, front: 10, 7 Sheet 4, back : 8, 9 Printing order for 4 pages: Sheet 1, front: 4, 1 Sheet 1, back : 2, 3

### Pregunta 4 (15 puntos)

La CONFECH, en su afán de agilizar el proceso de recuento de las votaciones, le ha encargado el desarrollo de un programa de registro de votación por universidades. Primero, el programa debe solicitar al usuario ingresar la cantidad de universidades que participan en el proceso. Luego, para cada una de las universidades, el usuario debe ingresar el nombre de la universidad y los votos de sus alumnos, que pueden ser: aceptar (A), rechazar (R), nulo (N) o blanco (B). El término de la votación se indica ingresando una X, tras lo cual se debe mostrar los totales de votos de la universidad, con el formato que se muestra en el ejemplo. Finalmente, el programa debe mostrar el resultado de la votación, indicando la cantidad de universidades que aceptan, que rechazan y en las que hubo empate entre estas dos opciones.

```

Número de universidades: 3
Universidad: USACH
Voto: A
Voto: R
Voto: A
Voto: N
Voto: X
USACH: 2 aceptan, 1 rechazan, 0 blancos, 1 nulos.
Universidad: UChile
Voto: A
Voto: B
Voto: A
Voto: X
UChile: 2 aceptan, 0 rechazan, 1 blancos, 0 nulos.
Universidad: PUC
Voto: A
Voto: R
Voto: R
Voto: A
Voto: X
PUC: 2 aceptan, 2 rechazan, 0 blancos, 0 nulos.
Universidades que aceptan: 2
Universidades que rechazan: 0
Universidades con empate: 1

```