Tel-102 Paralelo 1
2º Semestre 2020
Práctico 2
29/09

Instrucciones

- Lea con detenimiento cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con extensión **.cpp** con lo desarrollado. El nombre del archivo debe tener el siguiente formato:
  TEL102_P2_Nombre_Apellido.cpp, sin tíldes. (Ej. TEL102_P2_Mauricio_Araya.cpp)
- Enviar el archivo a través de la página de aula del ramo, sección "Práctico 2" hasta las 23:59:59 del día de mañana 30/09/2020 hora local continental de Chile (UTC-3).
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online**.
- Sea riguroso con las instrucciones de desarrollo.
- Comente adecuadamente el programa, describiendo lo que hace.
- ¡Éxito!

Cifrado Exótico

1. La empresa **ZOUM++**, importante compañía de videollamadas, ha sido ampliamente criticada por no cifrar sus contenidos transmitidos de forma adecuada. Por ello, ha decidido implementar un sistema de cifrado de base cuaternaria, esto es, ocupando una representación de la información utilizando 4 dígitos. El contenido está representado por secuencias de solo símbolos (i.e., $\{0, 1, 2, 3\}$), y el sistema de cifrado relaciona de la siguiente forma:

   - 0 se relaciona con 2, y viceversa.
   - 1 se relaciona con 3, y viceversa.

   Como en todo sistema de encriptación existe una llave que es conocida por todos los asistentes a la videollamada, y un mensaje a encriptar ocupando la llave. El sistema debe conectar los dígitos según las relaciones descritas arriba, tomando los dígitos de la llave *en orden* y buscando la primera aparición de su dígito de contraparte en el texto a encriptar. Mire las figuras de este documento para entender este proceso.

   ZOUM++ ha solicitado sus servicios como programador. Para eso le entrega la plantilla **p2.cpp** como base para su programa que se muestra a continuación. Los detalles del significado de las funciones será explicado más adelante.

lom.cpp

```
#include <iostream>
/* Practico 2
 * Las tildes han sido omitidas intencionalmente para evitar
 * problemas de compatibilidad
 */

struct cod{
    char digit;
    cod *connection;
};

/* Complete aqui su codigo para las funciones solicitadas */


int main()
{
    // Pruebe lo solicitado aca
    // Modifique los ejemplos para verificar otros casos.
    // Crear secuencias
    int l1=4; // Largo de llave de cifrado
    int l2=10; // Largo de texto a cifrar
    cod *h1 = createSequence(l1); // Llave de cifrado
    cod *h2 = createSequence(l2); // Texto a cifrar
    connectSequences(h1, l1, h2, l2); // Conecta dígitos entre secuencias
    showCRYPT(h2, l2); // Muestra texto cifrado

    return 0;
}
```

Note que la planilla contiene la estructura (`struct`) `cod`, que incluye un caracter `dígito` y un puntero a una estructura de tipo `cod`, vale decir, su relación de encriptado.

1. Cree la función `cod * createSequence(int length)` que recibe como entrada un entero. Esta crea un arreglo de tipo `cod` de largo `length` asociado a una secuencia y retorna un puntero de tipo `cod` que apunta a la primera posición del arreglo. La secuencia es solicitada al usuario por pantalla. Cada posición de la secuencia puede tomar solo valores `0,1,2 o 3`, como se muestra en el siguiente ejemplo.

TEL102_P2_Nombre_Apellido.cpp

```
int main(){
    ...
    int l1=4; // Largo de secuencia
    cod *h1 = createSequence(l1);
    ...
}
```

Salida (consola)

```
...
Creando Secuencia
Ingrese cat: 0
Ingrese cat: 3
Ingrese cat: 1
Ingrese cat: 1
...
```

2. Cree la función `void connectSequences(cod *h1, int l1, cod *h2, int l2)` que recibe como entrada dos arreglos de tipo `cod` como punteros (`h1` y `h2`) y sus respectivos largos (`l1` y `2`), y no tiene retorno. Se toma como base la secuencia `h1` y se recorre de forma ordenada cada uno de sus elementos para encontrar una dígito compatible en `h2`, también recorrida de forma ordenada. Al encontrar una pareja, conectar los elementos respectivos en `h1` y `h2` (usando el puntero `connection` desde el elemento de `h1`).

TEL102_P2_Nombre_Apellido.cpp

```
int main(){
    int l1=4; // Largo de la llave
    int l2=10; // Largo del texto
    cod *h1 = createSequence(l1); // Llave
    cod *h2 = createSequence(l2); // Texto
    connectSequences(h1, l1, h2, l2); //Conexion de digitos
    ...
}
```

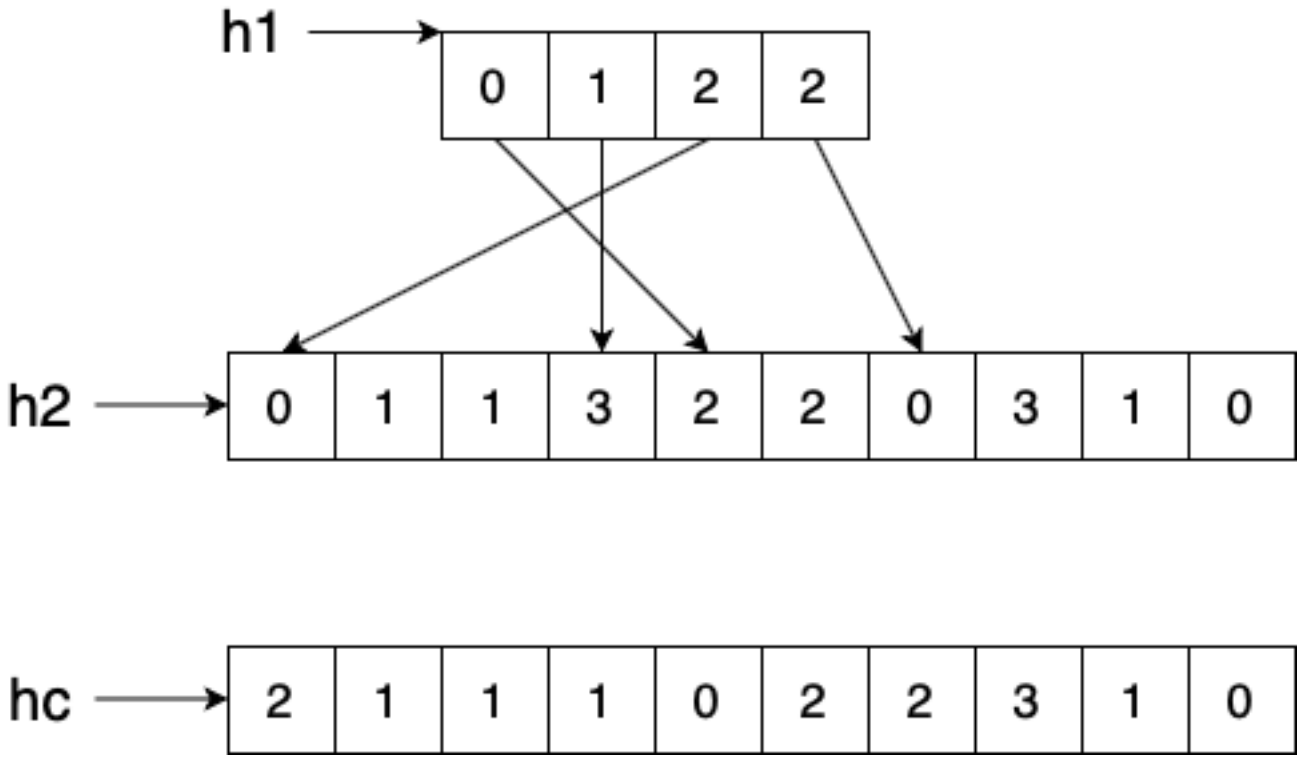

Figura 1: Esquema del resultado obtenido por **connectSequences()** para una secuencia *h2* dada.

3. Cree la función `showCRYPT(cod *h, int length)` que recibe como entrada un puntero de tipo `cod` al texto a encriptar `h` y su largo `length`, y que muestra por pantalla el texto `h` **reemplazando cada conexión por el dígito de la llave**. Guíese por el ejemplo.

Salida (consola)

```
foobar:practico2 mauricio$ ./crypt
Creando secuencia:
Ingrese valor
0
Ingrese valor
1
Ingrese valor
2
Ingrese valor
2
Creando secuencia:
Ingrese valor
0
Ingrese valor
1
Ingrese valor
1
Ingrese valor
3
Ingrese valor
2
Ingrese valor
2
Ingrese valor
0
Ingrese valor
3
Ingrese valor
1
Ingrese valor
0
La secuencia encriptada es:
2111022310
```