Guion de prácticas 3

Funciones y Arrays

DECSAI

Metodología de la Programación

Curso 2017/2018

Introducción al guion

En este guion se pondrán en práctica los conceptos asociados al uso de arrays y su paso como parámetros a funciones. Se construirán dos módulos que luego se integrarán en una biblioteca. La programación se realizará utilizando herramientas provistas en una instalación Linux estándar.

Módulo descifra

Suponga que un array de char representa un mensaje cifrado. La forma de descifrado consiste en coger la primera y última letra de cada palabra. Las palabras están separadas por uno o más espacios en blanco o el final del vector. Si la última palabra no tiene espacios en blanco a su derecha, se coge solo el primer carácter. Por ejemplo, si denotamos el inicio y final del mensaje con un corchete, entonces: `[ Hidrógeno limpia ]` se descifraría como `[Hola]`. Otros ejemplos son:

Ejemplo de entrada: `[Hidrógeno limpia ]` Salida correcta: `[Hola]`
Ejemplo de entrada: `[Hidrógeno limpia]` Salida correcta: `[Hol]`
Ejemplo de entrada: `[Hidrógeno]` Salida correcta: `[H]`
Ejemplo de entrada: `[Hidrógeno ]` Salida correcta: `[Ho]`
Ejemplo de entrada: `[H]` Salida correcta: `[H]`
Ejemplo de entrada: `[H ]` Salida correcta: `[H]`

Se pide implementar el módulo `descifra` que contenga las siguientes funciones.

1. **comienzaPalabra**: recibe un array de char v, su tamaño n y una posición j y devuelve verdadero o falso si en la posición j de v comienza o no una palabra.
2. **terminaPalabra**: recibe un array de char v, su tamaño n y una posición j y devuelve verdadero o falso si en la posición j de v termina o no una palabra.
3. **descifra**: recibe un array de char v, su tamaño n y devuelve un array rta que contiene el mensaje descifrado, así como su tamaño n_rta.
4. **toString**: recibe un array de char v, su tamaño n y devuelve un string que contiene los datos de v.

Módulo tiraLed

Un array de valores booleanos se puede utilizar para representar una tira de leds. Cada led estará encendido o apagado. Partiendo de esta idea, implemente un módulo `tiraLed` que contenga las siguientes funciones:

1. **encenderLed**: recibe un array de booleanos v, su tamaño n y una posición j, y "enciende" el led j.

2. **apagarLed**: ídem que el anterior, pero el led j se "apaga".
3. **cambiaLeds**: en lugar de una sola posición, se recibe un vector de enteros, y el tamaño de este, indicando las posiciones de los leds que van a cambiar de estado.
4. **toString**: recibe un array de booleanos v, su tamaño n y devuelve un string que contiene los datos de v.

Tareas a Realizar

Cree un directorio `practica3` que contenga a su vez los directorios `src, bin, include, lib, obj`. Luego

1. Implemente el módulo `descifra`.
2. Implemente el módulo `tiraLed`.
3. Construya una biblioteca `libpractica3.lib` que incluya ambos módulos
4. En un fichero `main.cpp` muestre ejemplos de uso de todas las funciones de la biblioteca. Al menos debe realizar las tareas siguientes. Para probar el primer módulo:
   a) Leer mensaje cifrado (suponga que el mensaje tiene como máximo 100 caracteres).
   b) Mostrar m En la Fig. 1 tiene un ejemplo de lectura.
   c) Descifrar el mensaje.
   d) Mostrar mensaje descifrado.

   Para probar el segundo módulo (suponga que la tira de led tiene como máximo 100 elementos):
   a) Apagar la tira.
   b) Mostrar la tira.
   c) Encender todos los leds de las posiciones pares.
   d) Mostrar la tira.
   e) Apagar todos los leds de las posiciones pares y encender los de las impares
   f) Mostrar la tira.
5. toda la compilación debe hacerse a través de un fichero `Makefile`

```
#include <iostream>
using namespace std;
const int MAX_SIZE = 100;

int main(){
char frase[MAX_SIZE];
int total_u = 0;
char letra;

cout << "Ingrese la frase. @ para terminar: ";
letra = cin.get();
while (letra != '@'){
frase[total_u] = letra;
total_u++;
letra = cin.get();
}

for(int i = 0; i < total_u; i++){
cout << frase[i];
}
cout << endl;

return(0);
}
```

Figura 1: Código para leer caracteres y almacenarlos en un vector.