*Introduction*

This report presents how we decode a Code-11 barcode including different cases, for instance , bar widths out of the allowable range, invalid start and stop , and detections of barcode errors as well. We will first introduce our software architecture , and then elaborate on how main functions work.

*the software design architecture*

1.讀取input的txt檔

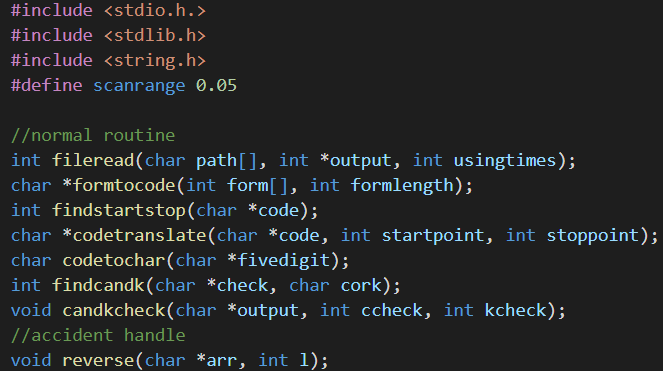
2.轉成一串1,0的code

3.translate成需要的char

4.產生c\_check和k\_check

5.判斷是否為badcode

例外處理:reverse code / out of range

*functions*

*Details*

**1.讀取input的txt檔**

Int scantimes=fileread(filepath,source,usingtimes);

(\*註:上式為程式執行時引用之function形式,其參數之格式可參照上方functions說明或原始碼變數宣告)

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 | (參數意義) | 回傳 | (回傳意義) |
| Filepath | 檔案路徑 | scantimes | 本次讀取數串數  =數串長度 |
| Source | 存取讀取數串 |  |  |
| usingtimes | 跨檔案讀取總數串數 |  |  |

內容:

1.用fscanf將數串以int傳出(source存取)

2.讀取前以fscanf讀取usingtimes次以跳過讀取過的區域

3.將這次讀取的次數回傳以便後續函式使用及累加至usingtimes

**2.轉成一串1,0的code**

Char \*code=formtocode(source,scantimes);

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 |  | 回傳 |  |
| source | fileread讀取到的數串 | \*code | 存取轉換碼(1,0)位址 |
| Scantimes | fileread的回傳值=數串長度 |  |  |

內容:

1.以\*source(數串第一個)為基準

2.若\*(source+i)位於其兩倍且考慮誤差值的區間

將其對應的output值設為2

3.若\*(source+i)位於其二分之一倍且考慮誤差值的區間

將其對應的output值設為0

4.若\*(source+i)位於其兩倍且考慮誤差值的區間

將其對應的output值設為1

5.將轉換到的字元陣列output回傳

**3.translate成需要的char**

Int startpoint=findstartstop(code);

Int stoppoint=findstartstop(startpoint+code+1)+startpoint+2;

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 |  | 回傳 |  |
| code | 已轉換的0,1碼 | Startpoint/stoppoint | 數串起始與結束的位置 |

內容:

用strstr在code裡面找到"00110"的字串，回傳其開頭位置是在code中的第幾位

Char \*output=codetranslate(code,startpoint,stoppoint);

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 |  | 回傳 |  |
| Code | 0,1碼 | \*output | 翻譯成character的轉換結果 |
| Startpoint/stoppoint | 算outputlength=  分組後(五個一組)的數串組數 |  |  |

內容:

1.將code從startpoint到stoppoint五個一組

2.傳入codetochar翻譯成對應character儲存到output陣列中

3.若無法翻譯則回傳NULL

**4.產生c\_check和k\_check**

findcandk(output,’c’)

findcandk(output,’k’)

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 |  | 回傳 |  |
| Output | 包含c\_check和k\_check的原字串 | Ccheck或kcheck |  |

內容:

將check去掉放置c\_check和k\_check的字元後，帶入公式得出c\_check和k\_check

**5.判斷是否為badcode**

candkcheck(output, findcandk(output,’c’);

candkcheck(output, findcandk(output,’k’);

|  |  |  |  |
| --- | --- | --- | --- |
| 引用參數 |  | 回傳 |  |
| output | 包含c\_check和k\_check的原字串 |  |  |
| findcandk(output,’c’) | c\_check &k\_check ,  用來判斷是否為Badcode |  |  |
| findcandk(output,’k’) |  |  |

內容:

判斷output的最後兩位是否分別對應到c\_check和k\_check

*work done by each student*

code: ALL

report

word: 柯映竹 ppt:洪宇辰

*comments*

During this project, the two most important thing that we’ve learned are:

1. How to solve a relatively complex problem.

Compared to those exercises we’ve done before, this project required more steps to solve the problem, and included more details that we had to think carefully about.

1. How to program with others.

This is the first time that we complete a program coding in a team. Since we are teammates, we have to communicate with each other, try to understand codes not only written by yourself, and the most important, make it easy for others to understand your codes.

From naming variables and functions , to writing comments in the program, it would be a lot easier for others to know what you’re doing in every line if a good name or comment is made.