

CS1003301

計算機程式設計 Computer Programming

Final Project

Code-11 Barcode Decoder

四資工一 B10915025 呂書瑋

四資工一 B10915033 王新翌

Introduction

這個專案是以可以接收barcode reader之輸入，且解碼code-11格式barcode為目的所做，由於沒有實際的環境，於是乎以text file作為模擬之資料流，此專案希望能在barcode reader可能發生的任何情況下保持正常運作，由於性質上仍然是模擬而不能確定運作的環境，所以相當程度上此專案僅止於紙上談兵，而不能臻至完美。

Algorithm

1. int scan(FILE* file, int* len, int num[])

此程式功能為從文件讀取測資並判斷是否有例外情況。

程式主要變數：

flag：紀錄資料是否有誤，初始值為0。

arr：暫存從檔暫中讀出的測資。

a_len：記錄每一筆測資的字元長度。

利用do while迴圈尋找第一個非空白字元，找到後將下一個空白前的所有字元存入arr中，並在最後加上NULL character，每次迴圈固定讀取一個數字(非一個位數)，並將此數字長度存在a_len中。

若a_len>3用於阻擋超長測資，以防錯誤測資進入atoi函式中引發錯誤。若

a_len<3則將測資由字元轉為數字型別存入len，len代表此筆資料的條碼數量。

接連兩個if分別為判斷len除以6的餘數是否等於5和len是否大於150，原因分別為條碼數量必須符合除以6等於5和條碼數量必須小於150才是有效測資，若有一條件不符和則flag++。

接下來利用從0~len-1的for迴圈將所有條碼寬度讀入arr，同樣利用a_len<3阻擋超長測資，a_len<3則將測資轉為數字型別存入num陣列，最後再判斷條碼寬度是否在1~200的範圍內，若超出範圍則flag++。

若最後flag不為0則代表讀取過程中有測資不符和規範，回傳-1；若所有測資皆符合規範則回傳0。

2. int sampling(int len, int num[], int exp[])

此程式功能為判斷所有條碼寬度是否皆有在誤差範圍內並將條碼轉為窄跟寬 (0 和 1) 兩種類別。

程式主要變數：

max：紀錄條碼最大寬度。

min：紀錄條碼最小寬度。

首先利用for迴圈將條碼寬度全部讀取一遍，並記錄下max和min，考量到最大寬度是200，最大的誤差值會是 $200 \times 0.05 = 10$ ，因此接下來for迴圈由min-10掃到max以確保掃到所有資料，若寬度在標準的0.95~1.05倍設為0，1.95~2.05倍設為1，若在此範圍中有數字能包住所有寬度則將此數字設為基準並回傳，否則回傳-1。

3. int decode(int len, int exp[], char str[])

此程式功能為判斷條碼讀取方向並將條碼轉為10進制數字。

程式主要變數：

tab：2進制條碼轉換表

syntax：10進制轉換表

dire：條碼方向，初始值為0

lc：由前往後讀計數器，初始值為0

rc：由後往前讀計數器，初始值為0

np：紀錄條碼陣列當前位置，初始值為0

bc：紀錄10進制陣列當前位置，初始值為0

flag和m_flag：紀錄條碼轉換過程中是否有誤，flag初始值為0

程式開始先利用for迴圈判斷程式開頭或結尾是否有符合由前往後或由後往前讀的規則，若由後往前讀則將dire設為0，若由前往後讀則將dire設為1。若兩者都不符合則回傳-1，若由後往前讀則將陣列倒轉。

接下來的while迴圈偵測0~陣列倒數第6個條碼，利用雙層for迴圈判斷陣列資料是否符合tab內的資料，若無符合則將flag設為1並跳出最內層迴圈；若符合則將由條碼轉換的數字存入str陣列中，並將m_flag++。

若m_flag為0代表此5個條碼無對應的數字可轉換，則此筆測資錯誤，回傳-1。

若m_flag不為0則判斷np+5是否為0，此數字為間隔字元，若成立則將np加6，否則回傳-1。

最後的for迴圈用於判斷最後5字元是否符合start/stop格式，若符合則將轉換成的結果存入str陣列中，並在最後加上NULL character，否則回傳-1。
若以上步驟皆無偵測到錯誤測資則回傳0。

4. int weight(char chr)

此程式功能為將條碼字元轉為權重並回傳。
利用switch case將字元轉換為對應的權重以用於c與k的計算。

5. int errcode(char* str)

此程式功能為檢查明碼中的偵錯碼C與K是否與計算出的偵錯碼相符合。
程式主要變數：

len：紀錄參數字串str的長度

c：紀錄str中的C偵錯碼

k：紀錄str中的K偵錯碼

此程式先於開頭以while迴圈紀錄str的長度，c與k則因為分別位於明碼的倒數第一位與第二位，又陣列index為0~len-1，於是取str[len-2]與str[len-3]為c與k。隨

即帶入公式 $\left(\sum_{i=1}^n ((n-i)\%10+1) \times w(c_i) \right) \% 11$ 與 $\left(\sum_{i=1}^{n+1} ((n-i+1)\%9+1) \times w(c_i) \right) \% 11$

並分別比對如若c或k錯誤則印出bad C或bad K，且回傳-1，如無發生意外則回傳0。

6. int main()

主程式，此程式的進入點，負責開檔、處理循序解碼過程、例外警示、輸出明碼。

程式主要變數：

time：紀錄現在的case數目

val：感測器的數量

num：感測器的資料

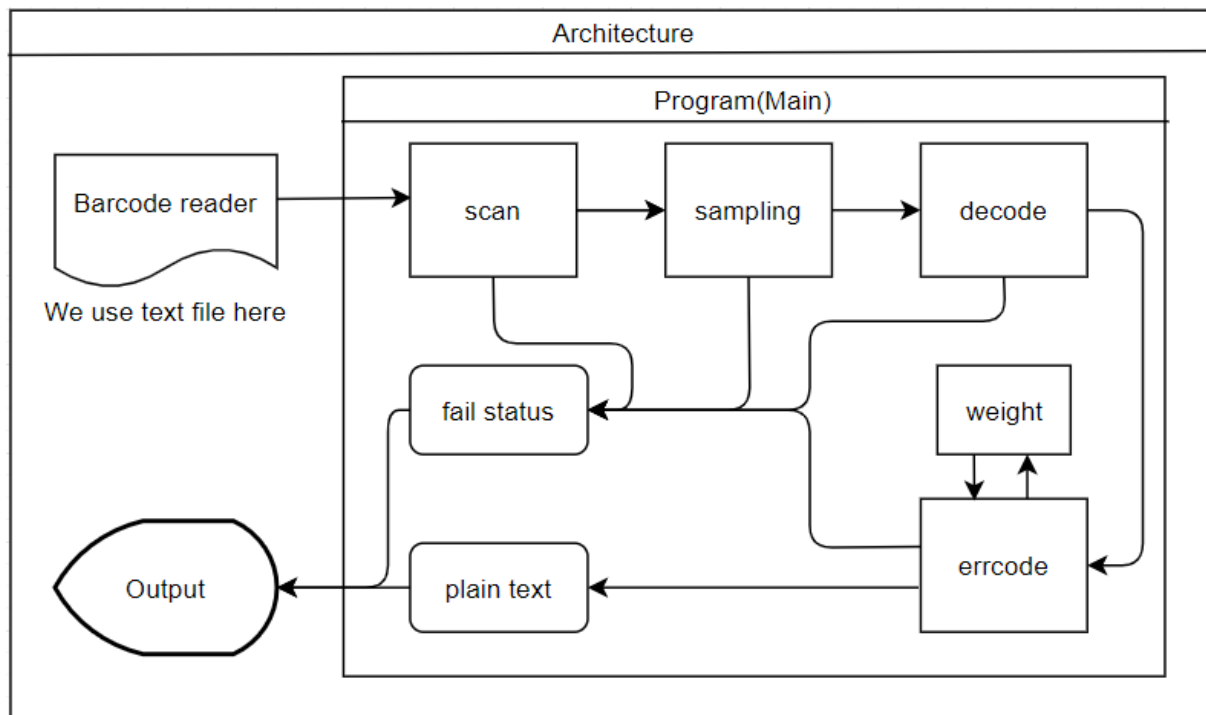
exp：經由sampling轉換過的資料分為寬和窄

scanResult、samplingResult、decodeResult：紀錄各步驟是否出現例外

主程式進入後先讀取檔名，隨後開檔並進入處理循環，呼叫scan函式如果得到val即感測器數量為0則return 0，結束此程式，否則呼叫sampling函式接著

decode函式，這過程中如果有函式回傳值不為0則印出bad code並進入下個循環，否則呼叫errcode函式回傳為0則印出最後得到的明文，如果回傳不為0則直接進入下個循環，進入下個循環前令time+1代表到下個case，至此這個程式運作所經歷的過程皆敘述完成。

Software Design Architecture



這個專案在預期下是要動作於barcode reader上，所以期望從硬體得到資料流，加上由於沒有確定的輸出硬體，我們預計程式的架構為，輸入硬體→軟體→輸出，軟體以函式為主構成。我們希望能從輸入硬體的資料流中抓取數值，經由scan函數得到數據，sampling函數量化barcode sensor數據的寬與窄，decode階段將bar code轉為明文，errcode檢查check character的運作，weight則是為了errcode轉換權重的函數。在轉換的過程中，如果有不符合barcode規則的情形發生，程式會進入fail status並其中可能包含了 bad code、bad C、bad K其中之一的情況，如果沒有發生錯誤，那程式進入plain text且準備好要輸出的正確明文，Output最後可以得到程式所處理完的結果。

Assignment

Both of the members have done an individual version, and we merge our program by considering every feature. Besides, the report is written by both of the students. We have pretty the same workload.

Comments

B10915025 呂書瑋：

我覺得這份專案的主題可以說是滿有趣的，但第一次以專案來看待算是讓我有點不知如何處理，如果是平常的程式題目如UVA、AMC上看到的那我可能考慮的範圍會比較好處理，當我希望這支程式能在大多情況下保持運作就有些問題浮上檯面，例如題目告訴我們感測器分區域檢測那會不會感測器有一些壞掉而沒有數值，由於輸入來自於text file而且也沒有嚴格規範，我們不得而知確切的每一次掃描間隔與哪裡，如果有真的barcode reader就能確定的對待每一筆case，在寫了幾種版本後最後還是採用折衷的作法，期待壞掉的感測器區域依然會回傳數值，兩人合作的過程也是一種新鮮的體驗，暴露出交換想法的能力需要加強，最後我想如果專案再大一點而且專案的使用環境可以再明確一點肯定會很有趣。

B10915033 王新翌：

這是我第一次與他人合作開發專案，過程中遭遇的最大問題是想法交流和整合。由於程式邏輯並不具體，要以人類語言完整表達十分困難，再加上在整合程式前我們各自皆有寫出一份功能完整的程式，但兩人寫出的程式運作邏輯不盡相同，因此在最後整合時須不斷比較兩份程式的差異並從中找出漏洞，才能產出功能完善的程式。此外，我們兩人命名變數的方式也有差異，這使我們在校閱彼此程式時需要不斷與對方確認變數功能，也在過程中學到許多有趣的命名規則，這也是多人開發專案的有趣之處，也使我體會到多人開發專案遠比多人做報告難上許多，做報告只需負責自己的部分，而開發專案須顧及到變數名稱是否重複、更改變數內容的動作是否會影響到後面程式的執行等等。藉由這個機會讓我有合作開發專案的體驗，相信未來若遇到類似的開發模式必定能更加得心應手。

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <ctype.h>
int scan(FILE* file,int* len,int num[]){
    int flag = 0;
    char arr[99];
    int a_len=1;
    do{
        fscanf(file,"%c",&arr[0]);
    }while(isspace(arr[0]));
    while(1){
        fscanf(file,"%c",&arr[a_len]);
        if(isspace(arr[a_len])){
            arr[a_len] = '\0';
            break;
        }
        a_len++;
    }
    if(a_len>3){
        flag++;
    }else{
        *len = atoi(arr);
    }
    if(*len%6!=5){
        flag++;
    }
    if(*len>150){
        flag++;
    }
    int i;
    for(i=0;i<*len;i++){
        a_len=1;
        do{
            fscanf(file,"%c",&arr[0]);
        }while(isspace(arr[0]));
        while(1){
            fscanf(file,"%c",&arr[a_len]);
            if(isspace(arr[a_len])){
                arr[a_len] = '\0';
                break;
            }
            a_len++;
        }
        if(a_len>3){
```

```

        flag++;
    }else{
        num[i] = atoi(arr);
    }
    if(num[i]>200||num[i]<1){
        flag++;
    }
}
if(flag){
    return -1;
}
return 0;
}
int sampling(int len,int num[],int exp[]){
    int i,max=0,min=9999;
    for(i=0;i<len;i++){
        if(num[i]>max)
            max = num[i];
    }
    min = max;
    for(i=0;i<len;i++){
        if(num[i]<min)
            min = num[i];
    }
    int j;
    for(i=min-(len*0.1);i<=max;i++){
        double l1 = (double)i*0.95;
        double l2 = (double)i*1.05;
        double u1 = (double)i*2*0.95;
        double u2 = (double)i*2*1.05;
        int count = 0;
        for(j=0;j<len;j++){
            if(l1<=num[j]&&l2>=num[j]){
                exp[j] = 0;
            }else if(u1<=num[j]&&u2>=num[j]){
                exp[j] = 1;
            }else{
                count++;
                break;
            }
        }
        if(count==0){
            return 0;
        }
    }
    return -1;
}
int decode(int len,int exp[],char str[]){

```



```

int tab[12][5] = {{0,0,0,1,1},{0,1,0,0,1},{1,0,0,0,0},
                  {1,0,1,0,0},{1,1,0,0,0},{0,0,1,0,1},
                  {0,1,1,0,0},{0,0,0,0,1},{1,0,0,1,0},
                  {1,0,0,0,1},{0,0,1,0,0},{0,0,1,1,0}};
char syntax[12] = {'0','1','2','3','4','5','6','7','8','9','-','#'};
int dire = 0;
int i,lc=0,rc=0;
for(i=0;i<5;i++){
    if(exp[i] == tab[11][i])
        lc++;
}
if(lc==5)
    dire = 0;
for(i=0;i<5;i++){
    if(exp[len-1-i] == tab[11][i])
        rc++;
}
if(rc==5)
    dire = 1;
if(lc!=5&&rc!=5){
    return -1;
}
if(dire==1){
    int tmp[200];
    for(i=0;i<len;i++){
        tmp[i] = exp[len-1-i];
    }
    for(i=0;i<len;i++){
        exp[i] = tmp[i];
    }
}
int np = 0;
int bc = 0;
while(len-np>5){
    int j,flag,m_flag=0;
    for(i=0;i<12;i++){
        flag=0;
        for(j=0;j<5;j++){
            if(exp[np+j]!=tab[i][j]){
                flag=1;
                break;
            }
        }
        if(flag)
            continue;
        else{
            str[bc++] = syntax[i];
            m_flag++;
        }
    }
}

```

```

    }
    if(!m_flag){
        return -1;
    }
    if(exp[np+5]==0){
        np+=6;
    }else{
        return -1;
    }
}
int j,flag,m_flag=0;
for(i=0;i<12;i++){
    flag=0;
    for(j=0;j<5;j++){
        if(exp[np+j]!=tab[i][j]){
            flag=1;
            break;
        }
    }
    if(flag)
        continue;
    else{
        str[bc++] = syntax[i];
        str[bc++] = '\0';
        np+=6;
        m_flag++;
    }
}
if(!m_flag){
    return -1;
}

return 0;
}
int weight(char chr){
    switch(chr){
        case '0':
            return 0;
        case '1':
            return 1;
        case '2':
            return 2;
        case '3':
            return 3;
        case '4':
            return 4;
        case '5':
            return 5;
        case '6':

```

```

        return 6;
    case '7':
        return 7;
    case '8':
        return 8;
    case '9':
        return 9;
    case '-':
        return 10;
    }
}
int errcode(char* str){
    int len = 0;
    while(str[len]!='\0'){
        len++;
    }
    char c = str[len-3];
    char k = str[len-2];
    int i,sum=0;
    for(i=1;i<len-3;i++){
        sum+=((len-4-i)%10+1)*weight(str[i]);
    }
    sum%=11;
    if(sum!=weight(c)){
        printf("bad C\n");
        return -1;
    }
    sum=0;
    for(i=1;i<len-2;i++){
        sum+=((len-3-i)%9+1)*weight(str[i]);
    }
    sum%=11;
    if(sum!=weight(k)){
        printf("bad K\n");
        return -1;
    }
    return 0;
}
int main(){
    int time=1;
    FILE * file;
    char str[999];
    printf("Enter file name:");
    scanf("%s",str);
    file = fopen(str,"r");
    while(1){
        int num[200] = {0};
        int exp[200] = {0};
        char str[200] = {'\0'};
        int val;

```

```

        int i;
        int scanResult = scan(file,&val,num);
        if(val==0){
            return 0;
        }
        printf("Case %d: ",time);
        if(scanResult==0){
            int samplingResult = sampling(val,num,exp);
            if(samplingResult==0){
                int decodeResult = decode(val,exp,str);
                if(decodeResult==0){
                    if(errcode(str)==0){
                        int len = 0;
                        while(str[len]!='\0'){
                            len++;
                        }
                        for(i=1;i<len-3;i++){
                            printf("%c",str[i]);
                            printf("\n");
                        }
                    }
                    else{
                        printf("bad code\n");
                    }
                }
            }
            else{
                printf("bad code\n");
            }
        }
        else{
            printf("bad code\n");
        }
        time++;
    }
    return 0;
}

```