|  |
| --- |
| 得 分 |
|  |

資料結構

HWK

KMP收尋

B10702057盧昱達

* 程式碼

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

#include<windows.h>

#define SIZE 2000

int input\_file(char input[])

{

int i = 0;

FILE \*fp;

char filename[20];

printf("請輸入檔名:");

scanf("%s", filename);

fp = fopen(filename, "r");

if (fp == NULL)

{

printf("開檔失敗囉!!\n");

return -1;

}

else

{

while(!feof(fp))

{

fscanf(fp, "%c", &input[i]);

i++;

}

}

fclose(fp);

return i;

}

void LoseF(char pat[], int pat\_Lose\_F[])

{

int length = strlen(pat);

pat\_Lose\_F[0] = -1;

int start = 1,i,j;

for(i = 2;i <= (length); i++)

{

int start = 1;

int current = 0;

for(j = start; j < i; j++)

{

if( pat[j] != pat[current++])

{

current = 0;

j = start++;

}

}

if( start < i )

{

pat\_Lose\_F[i-1] = current - 1;

}

else

{

pat\_Lose\_F[i-1] = -1;

}

}

}

int KMPseach(char pat[], char str[], int out[])

{

int total = 0;

int Patlen = strlen(pat);

int Strlen = strlen(str);

int Fail[Patlen], i = 0, j = 0, n = 0;

LoseF(pat, Fail);

for (i = 0; i < Strlen;)

{

if (str[i] != pat[j])

{

if (j == 0)

{

i++;

}

else

{

j = Fail[j - 1] + 1;

}

}

else

{

i++;

j++;

}

if (j == (Patlen - 1))

{

out[n] = i - j;

j = 0;

n++;

}

total++;

}

if (j < Patlen && n == 0)

{

return -1;

}

else

{

return total;

}

}

void setColor(int color)

{

HANDLE hConsole;

hConsole = GetStdHandle (STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, color);

}

void printOUT(char str[], char pat[], int out[])

{

int out\_n, str\_n, pat\_n;

int i, k = 0;

out\_n = sizeof(out) / sizeof(out[0]);

str\_n = strlen(str);

pat\_n = strlen(pat);

printf("======== 收尋結果 ========\n");

for(i = 0; i < str\_n; i++)

{

if(i == out[k] && i < out[k] + pat\_n)

{

for ( int p = 0; p < pat\_n; p++)

{

setColor(12);

printf("%c", str[i]);

setColor(15);

i++;

}

i--;

k++;

}

else

printf("%c", str[i]);

}

printf("==========================\n");

}

int main ( void )

{

int str\_i, pat\_n;

int total;

int pat\_Lose\_F[30];

int out[SIZE] = {0};

char str[SIZE], pat[30];

int seach\_n = 0;

//讀檔

str\_i = input\_file(str);

printf("%s\n", str);

printf("全文有%d字元", str\_i);

printf("請輸入要收尋的字串(30字元內): ");

scanf("%s", pat);

while(pat != NULL)

{

//讀入收尋字串

pat\_n = strlen(pat);

//失敗函數

LoseF(pat, pat\_Lose\_F);

printf("%s失敗函數為:\n", pat);

for ( int i = 0; i < pat\_n; i++) printf("%d ", pat\_Lose\_F[i]);

printf("\n");

//KMP收尋

total = KMPseach(pat, str, out);

if (total == -1)

{

printf("此文無%s\n", pat);

printf("%s", str);

}

else

{

printf("總共收尋了%d次\n", total);

printOUT(str, pat, out);

}

//列印結果

printf("請輸入要收尋的字串(30字元內): ");

scanf("%s", pat);

}

system("pause");

}

* 測試結果



* 實作心得

在這次實作功課時，在寫失敗函數中，我發現我對C&C++有需多小細節不熟，在很多地方常會出現小錯誤，導致寫不出理想中的答案，或是一直卡關，還有在思考如何產生失敗函數時，發生許多小問題，最後需要靠著GOOGLE，與老師的講解，才能解出答案。