作業題目：Linking loader

作法：

預設開始位址為 1000

若欄位 1 是 H 的話會記錄 control section 、計算過的位址、長度到 loadmap。

若欄位 2 是 T 的話會記錄 symbol name 跟計算過的位址到 loadmap。

Ps：位址會不斷累加。

原程式碼：

```
<stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <iostream>
#include <cstdlib>
using namespace std;
char* DecToHex(int dec);
int HexToDec(const char* hexPoint);
char* stringAdd(const char* , const char* );
char* addZero(const char*, int );

int main(void)
{
	int mapindex = 0;
	string startAddress = "1000";
	string loadMap[5][4];
	string proga[4][10] = {{"H","^","PROGA","^","000000","^","0082"}
				,{"D","^","LISTA","^","000036","^","ENDA","^","00002
4"}
				,{"T","^","000000","^","03","^","000011",}
```

```
                                    ,{"E","^","000000"}};

    string progb[3][10] = {{"H","^","PROGB","^","000000","^","0055"}
                                    ,{"D","^","LISTB","^","000033"}
                                    ,{"E","^","000000"}};
    string top[4] = {"Control section","Symbol name","Address","Length"};
    for(int i=0; i<4; i++){

        if(proga[i][0] == "H"){

            loadMap[mapindex][0] = proga[i][2];
            loadMap[mapindex][2] = DecToHex(HexToDec(proga[i][4].c_str()) +
HexToDec(startAddress.c_str()));
            loadMap[mapindex][3] = DecToHex(HexToDec(proga[i][6].c_str()));
            mapindex++;
        }
        if(proga[i][0] == "D"){

            loadMap[mapindex][1] = proga[i][2];
            loadMap[mapindex][2] =
DecToHex(HexToDec(proga[i][4].c_str())+HexToDec(startAddress.c_str()));
            mapindex++;
            loadMap[mapindex][1] = proga[i][6];
            loadMap[mapindex][2] =
DecToHex(HexToDec(proga[i][8].c_str())+HexToDec(loadMap[mapindex-
1][2].c_str()));
            mapindex++;
        }
    }
    for(int i=0; i<3; i++){

        if(proga[i][0] == "H"){

            loadMap[mapindex][0] = progb[i][2];
            loadMap[mapindex][2] = DecToHex(HexToDec(progb[i][4].c_str()) +
HexToDec(loadMap[0][3].c_str())+HexToDec(startAddress.c_str()));
            loadMap[mapindex][3] = DecToHex(HexToDec(progb[i][6].c_str()));
            mapindex++;
```

```
            }
            if(proga[i][0] == "D"){

                loadMap[mapindex][1] = progb[i][2];
                loadMap[mapindex][2] = DecToHex(HexToDec(progb[i][4].c_str()) +
HexToDec(loadMap[mapindex-1][2].c_str()));
                mapindex++;
            }
        }
        printf("A Load Map:\n\n");
        printf("%-20s%-20s%-20s%-20s",top[0].c_str(), top[1].c_str(), top[2].c_str(),
top[3].c_str());
        printf("\n----------------------------------------------------------------\n");
        for(int i=0; i<5; i++){

            for(int j=0; j<4; j++){

                printf("%-20s",loadMap[i][j].c_str());
            }
            printf("\n");
        }



        return 0;
    }
    // 幫字串補零
    char* addZero(const char* s_in, int length)
{
        string zero0 = "", zero1 = "0", zero2 = "00", zero3 = "000", zero4 = "0000", zero5
= "00000";
        char output[16];
        if(6 - length == 5){
            return stringAdd(zero5.c_str(), s_in);
        }
        else if(6 - length == 4){
            return stringAdd(zero4.c_str(), s_in);
        }
        else if(6 - length == 3){
```

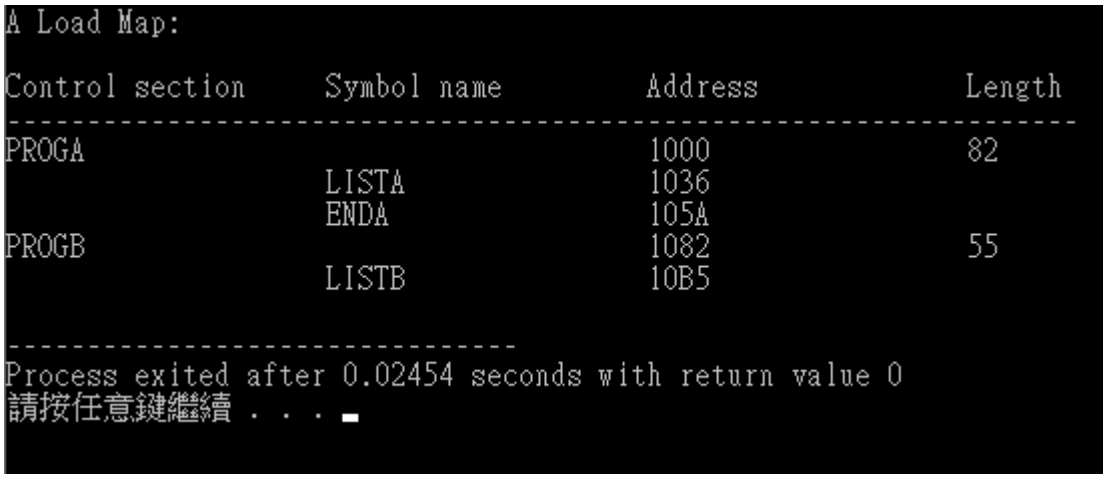
```
            return stringAdd(zero3.c_str(), s_in);
        }
        else if(6 - length == 2){
            return stringAdd(zero2.c_str(), s_in);
        }
        else if(6 - length == 1){
            return stringAdd(zero1.c_str(), s_in);
        }
        else
           return stringAdd(zero0.c_str(), s_in);;
     }
//字串相連
char* stringAdd(const char* sa, const char* sb)
{
        char *a;
        char *b;
        char final[9];
        a=const_cast<char*>(sa); //consr char* To char*
        b=const_cast<char*>(sb);
        strcat(a,b); //字串相連
        return a;
}
//10 進制 轉 16 進制
char* DecToHex(int dec)
{
        char Hex[9];
        sprintf(Hex, "%X", dec);
        return Hex;
}
//16 進制 轉 10 進制
int HexToDec(const char* hexPoint)
{
        char hex[9];
        strncpy(hex, hexPoint, strlen(hexPoint) + 1);
        long long decimal, place;
        int i = 0, val, len;

        decimal = 0;
```

```
    place = 1;

    len = strlen(hex);
    len--;

    for(i=0; hex[i]!='\0'; i++){

    // Find the decimal representation of hex[i]
    if(hex[i]>='0' && hex[i]<='9')
    {
    val = hex[i] - 48;
    }
    else if(hex[i]>='a' && hex[i]<='f')
    {
    val = hex[i] - 97 + 10;
    }
    else if(hex[i]>='A' && hex[i]<='F')
    {
    val = hex[i] - 65 + 10;
    }

    decimal += val * pow(16, len);
    len--;
    }

    return decimal;
}
```

測試資料如下圖：

```
string proga[4][10] = {{"H","^","PROGA","^","000000","^","0082"}
                      ,{"D","^","LISTA","^","000036","^","ENDA","^","000024"}
                      ,{"T","^","000000","^","03","^","000011",}
                      ,{"E","^","000000"}};

string progb[3][10] = {{"H","^","PROGB","^","000000","^","0055"}
                      ,{"D","^","LISTB","^","000033"}
                      ,{"E","^","000000"}};
```

結果輸出如下圖：



討論：輸出到記憶體的部分有點困難，因此沒有全做出來。

心得：有空閒的話會嘗試檔案輸入，而不是直接打在程式碼。