程設hw04 40947046S 劉佩昀

hw0401

1. ./hw0401
2. 輸入檔名實不要忘記輸副檔名。
3. 以下為程式介紹

在此題中，由於每個pixel要放大四倍，所已我寫了四個function去填放大後pixel的值，

```
void black_1(pixel* a,pixel* b,pixel* c,pixel*d){
    a->Blue=255;
    a->Green=255;
    a->Red=255;
    b->Blue=0;
    b->Green=0;
    b->Red=0;
    c->Blue=0;
    c->Green=0;
    c->Red=0;
    d->Blue=255;
    d->Green=255;
    d->Red=255;
}

void black_2(pixel* a,pixel* b,pixel* c,pixel*d){
    a->Blue=0;
    a->Green=0;
    a->Red=0;
    b->Blue=255;
    b->Green=255;
    b->Red=255;
    c->Blue=255;
    c->Green=255;
    c->Red=255;
    d->Blue=0;
    d->Green=0;
    d->Red=0;
}

void white(pixel* a,pixel* b,pixel* c,pixel*d){
    a->Blue=255;
    a->Green=255;
    a->Red=255;
    b->Blue=0;
    b->Green=0;
    b->Red=0;
    c->Blue=0;
    c->Green=0;
    c->Red=0;
    d->Blue=255;
    d->Green=255;
    d->Red=255;
}

```

```
void gray(pixel* a,pixel* b,pixel* c,pixel* d,uint8_t color){
    if(color>127){
        a->Blue=255;
        a->Green=255;
        a->Red=255;
        b->Blue=255;
        b->Green=255;
        b->Red=255;
        c->Blue=255;
        c->Green=255;
        c->Red=255;
        d->Blue=255;
        d->Green=255;
        d->Red=255;
    }else{
        a->Blue=0;
        a->Green=0;
        a->Red=0;
        b->Blue=0;
        b->Green=0;
        b->Red=0;
        c->Blue=0;
        c->Green=0;
        c->Red=0;
        d->Blue=0;
        d->Green=0;
        d->Red=0;
    }
}
```

hw0402

1. ./hw0402
2. 使用fprintf把文字寫入新的.h檔和.c檔

hw0403

1. ./hw0403
2. code一行不能超過1024，不然我的buffer會爆掉。
3. 檔名請放在最後一個argument
4. 以下為程式介紹：

使用getopt_long

```
   struct option long_options[] = {
     { "linenum" , 0, NULL, 'n' },
     { "color"   , 0, NULL, 'c' },
     { "language", 1, NULL, 'l' },
     {  0        , 0,  0  ,  0  },               //規定：最後一定要有這行，表示結束，這是man的規定。
};
```

由於題目有兩種選擇：上色或加行數
所以我分成上色、加行數、上色且加行數，這三個function。
至於比對關鍵字的部分寫死在code裡，長這樣：

```
const char *rust_language[]={
    "as"      , "break" , "const"   , "continue",
    "crate"   , "else"  , "enum"    , "extern",
    "false"   , "fn"    , "for"     , "if",
    "impl"    , "in"    , "let"     , "loop",
    "match"   , "mod"   , "move"    , "mut",
    "pub"     , "ref"   , "return"  , "self",
    "Self"    , "static", "struct"  , "super",
    "trait"   , "true"  , "type"    , "unsafe"
    "use"     , "where" , "while"   , "async",
    "await"   , "dyn"   , "abstract", "become",
    "box"     , "do"    , "final"   , "macro",
    "override", "priv"  , "typeof"  , "unsized",
    "virtual" , "yeild" , "try"     , "union",
    "'static"
};
const char *java_language[]={
    "abstract", "assert"      , "boolean" , "break",
    "byte"    , "case"        , "catch"   , "char",
    "class"   , "const"       , "continue", "defalt",
    "do"      , "double"      , "else"    , "enum",
    "extends" , "final"       , "finally" , "float",
    "for"     , "if"          , "goto"    , "implements",
    "import"  , "instanceof"  , "int"     , "interface",
    "long"    , "native"      , "new"     , "package",
    "private" , "protected"   , "public"  , "return",
    "short"   , "static"      , "strictfp", "super",
    "switch"  , "synchronized", "this"    , "throw",
    "throws"  , "transient"   , "try"     , "void",
    "volatile", "while"       , "_"
};

const char *js_language[]={
    "await"  ,"break" ,"case"      , "catch"   ,
    "class"  ,"const" ,"continue"  , "debugger",
    "default","delete","do"        , "else"    ,
    "enum"   ,"export","extends"   , "false"   ,
    "finally","for"   ,"function"  , "if"      ,
    "import" ,"in"    ,"instanceof", "new"     ,
    "null"   ,"return","super"     , "switch"  ,
    "this"   ,"throw" ,"true"      , "try"     ,
    "typeof" ,"var"   ,"void"      , "while"   ,
    "with"   ,"yield"
};

const char *c_plus_language[]={
    "alignas"      , "alignof"     , "asm"             , "auto"     ,
    "bool"         , "break"       , "case"            , "catch"    ,
    "char"         , "char16_t"    , "char32_t"        , "class"    ,
    "const"        , "constexpr"   , "const_case"      , "continue" ,
    "decltype"     , "default"     , "delete"          , "do"       ,
    "double"       , "dynamic_case", "else"            , "enum"     ,
    "explicit"     , "export"      , "extern"          , "false"    ,
    "float"        , "for"         , "friend"          , "goto"     ,
    "if"           , "inline"      , "int"             , "long"     ,
    "mutable"      , "namespace"   , "new"             , "noexcept" ,
    "nullptr"      , "operator"    , "private"         , "protected",
    "public"       , "register"    , "reinterpret_cast", "return"   ,
    "short"        , "signed"       , "sizeof"           , "static"   ,
    "static_assert", "static_cast"  , "struct"           , "switch"   ,
    "template"     , "this"         , "thread_local"     , "throw"    ,
    "true"         , "try"          , "typedef"          , "typeid"   ,
    "typename"     , "union"        , "unsigned"         , "using"    ,
    "virtual"      , "void"         , "volatile"         , "wchar_t"  ,
    "while"        , "and"          , "and_eq"           , "bitand"   ,
    "bitor"        , "compl"        , "not"              , "not_eq"   ,
    "or"           , "or_eq"        , "xor"              , "xor_eq"   ,
};

const char *C_language[]={
    "auto"       , "break"      , "case"             , "char"          ,
    "const"      , "continue"   , "default"          , "do"            ,
    "double"     , "else"       , "enum"             , "extern"        ,
    "float"      , "for"        , "goto"             , "if"            ,
    "inline"     , "int"        , "long"             , "register"      ,
    "restrict"   , "return"     , "short"            , "signed"        ,
    "sizeof"     , "static"     , "struct"           , "switch"        ,
    "typedef"    , "union"      , "unsigned"         , "void"          ,
    "volatile"   , "while"      , "_Alignas"         , "_Alignof"      ,
    "_Atomic"    , "_Bool"      , "_Complex"         , "_Generic"      ,
    "_lmaginary" , "_Noreturn"  , "_Static_assert"   , "Thread_local"
};
```

hw0404

1. ./hw0404
2. 能做到修改attribute、金錢、健康狀況