```
    #!/bin/bash
    #
    # Author: Michael Kepple
    # Date: 24 Jan 2013
    #
    # This script performs syntax highlighting for the C language.
    #
    saveIFS=$IFS
    while read line; do
    IFS=" "
    words=(${line})
    IFS=$saveIFS
    len=${#words[@]};
    for (( i=0; i<len; i++ ))
    do
    if [[ "${words[$i]}" == "auto" || "${words[$i]}" == "break" ||
         "${words[$i]}" == "case" || "${words[$i]}" == "char" ||
         "${words[$i]}" == "const" || "${words[$i]}" == "continue" ||
        "${words[$i]}" == "default" || "${words[$i]}" == "do" ||
        "${words[$i]}" == "double" || "${words[$i]}" == "else" ||
        "${words[$i]}" == "enum" || "${words[$i]}" == "extern" ||
        "${words[$i]}" == "float" || "${words[$i]}" == "for" ||
        "${words[$i]}" == "goto" || "${words[$i]}" == "if" ||
        "${words[$i]}" == "int" || "${words[$i]}" == "long" ||
        "${words[$i]}" == "register" ||"${words[$i]}" == "return" ||
         "${words[$i]}" == "short" || "${words[$i]}" == "signed" ||
          "${words[$i]}" == "sizeof" || "${words[$i]}" == "static" ||
          "${words[$i]}" == "struct" || "${words[$i]}" == "switch" ||
          "${words[$i]}" == "typedef" || "${words[$i]}" == "union" ||
          "${words[$i]}" == "unsigned" || "${words[$i]}" == "void" ||
          "${words[$i]}" == "volatile" ||"${words[$i]}" == "while" ]]
    then
    printf "*${words[$i]}* "
    else
        echo "${words[$i]}" | sed 's/^[a-zA-Z_][a-zA-Z_0-9]*/^&^/' |
        sed "s/\([^\^]*[[ &!\"\'\*][[ &!\"\'\*]*\)\([a-zA-Z_][a-zA-Z_0-9]*\)\([^\^]*\)*
    /\1^\2^\3/g" | tr '\n' ' '
    fi
    done
    printf "\n";
    done < $1

    :'
    ================================================================================
    Test
    ================================================================================
    /
     / ^http^://www.programmingsimplified.com/c/source-code/c-program-selection-sort
     /

     include<stdio.h>

     main^()

     int* ^array^[100], ^n^, ^c^, ^d^, ^position^, ^swap^;

     printf^("^Enter^ ^number^ ^of^ ^elementsn^");
     scanf^("%d", &^n^);

     printf^("^Enter^ %d ^integersn^", ^n^);

     for* ( ^c^ = 0 ; ^c^ < ^n^ ; ^c^++ )
     scanf^("%d", &array[^c^]);
```

```
     for* ( ^c^ = 0 ; ^c^ < ( ^n^ - 1 ) ; ^c^++ )

     position^ = ^c^;

     for* ( ^d^ = ^c^ + 1 ; ^d^ < ^n^ ; ^d^++ )

     if* ( ^array^[^position^] > ^array^[^d^] )
     position^ = ^d^;

     if* ( ^position^ != ^c^ )

     swap^ = ^array^[^c^];
     array^[^c^] = ^array^[^position^];
     array^[^position^] = ^swap^;



     printf^("^Sorted^ ^list^ ^in^ ^ascending^ ^order^:n");

     for* ( ^c^ = 0 ; ^c^ < ^n^ ; ^c^++ )
     printf^("%dn", ^array^[^c^]);

     return* 0;
```