

```
; Attributes: bp-based frame fuzzy-sp  
;  
; int __cdecl main(int argc, const char **argv, const char **envp)  
public main  
main proc near  
  
x= dword ptr -10h  
var_C= dword ptr -0Ch  
var_4= dword ptr -4  
argc= dword ptr 8  
argv= dword ptr 0Ch  
envp= dword ptr 10h  
  
lea      ecx, [esp+4]  
and     esp, 0FFFFFFF0h  
push    dword ptr [ecx-4]  
push    ebp  
mov     ebp, esp  
push    ecx  
sub     esp, 14h  
call    __x86_get_pc_thunk_ax  
add     eax, 1A77h  
mov     [ebp+var_C], 7  
mov     [ebp+x], 0  
mov     [ebp+x], 1  
jmp     short loc_58F
```

The diagram illustrates the control flow from the `main proc near` section to the `loc_58F:` label. A blue arrow points down from the assembly code to the `loc_58F:` label. Another blue arrow originates from the `jmp short loc_58F` instruction and points to the `loc_58F:` label. A red arrow originates from the `jle short loc_57D` instruction and points to the `loc_57D:` label.

```
loc_58F:  
cmp     [ebp+x], 0Eh  
jle     short loc_57D
```

A green arrow points down from the `loc_58F:` label to the `loc_57D:` label.

```
loc_57D:  
sub    esp, 0Ch  
push    [ebp+x]          ; x  
call    printFunction  
add    esp, 10h  
add    [ebp+x], 1
```

A red arrow points right from the `loc_57D:` label to the `main endp` label. A blue arrow points down from the `loc_57D:` label to the `main endp` label.

```
nop  
mov     ecx, [ebp+var_4]  
leave  
lea     esp, [ecx-4]  
retn  
main endp
```