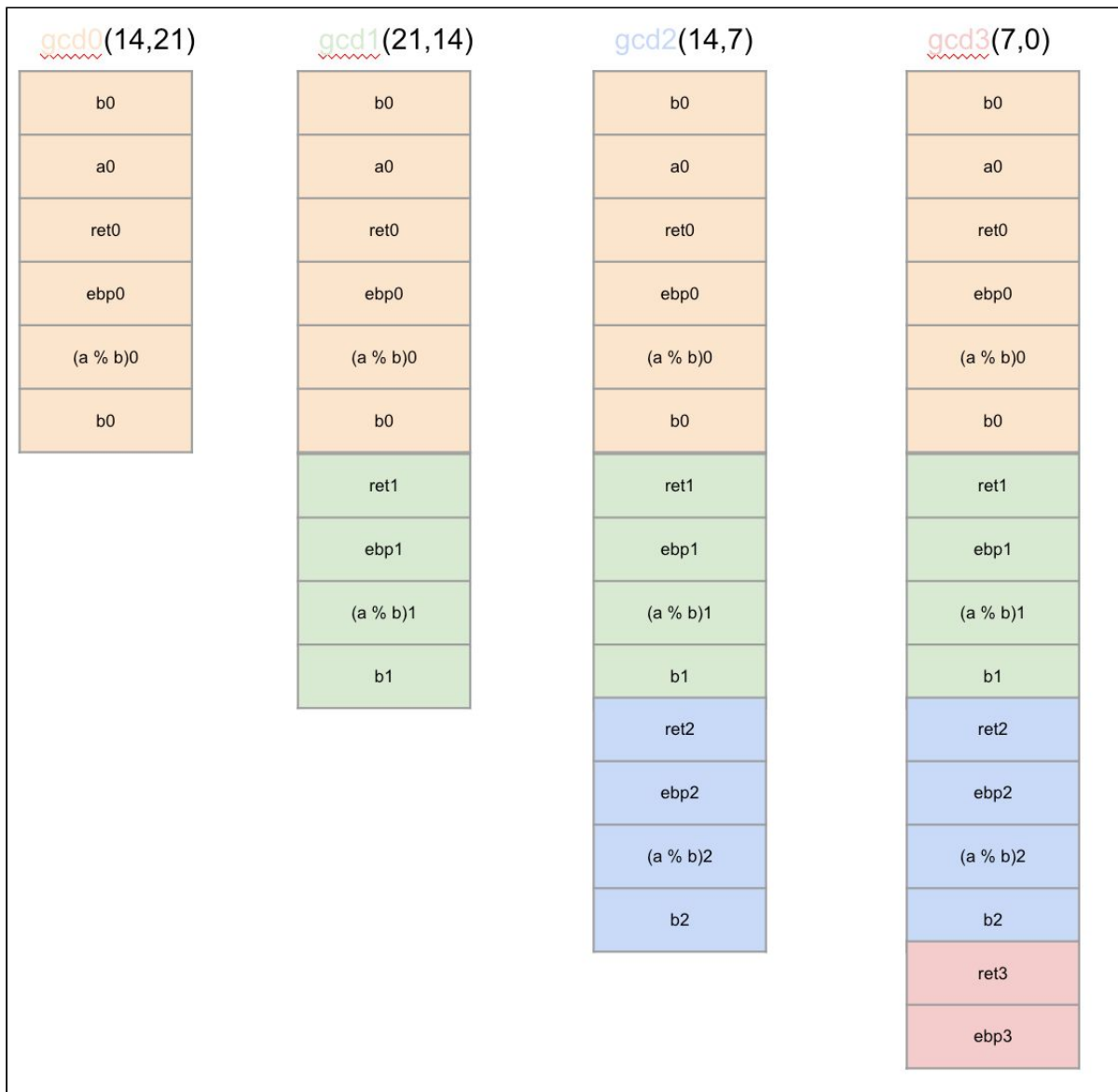


## CS3021 Computer Architecture II - Tutorial 1

*Student # 16327446 - Brandon Dooley*

2)

**Maximum Stack Depth (in Frames) = 4** (See diagram below)



### 3) Code

[t1.asm] - min

```
1  .686                      ; create 32 bit code
2  .model flat, C            ; 32 bit memory model
3  option casemap:none       ; case sensitive
4
5  .data                     ; start of data section
6  public g                  ; export of variable g
7  g DWORD 4                 ; declare global variable g initialised to 4
8
9  .code                     ; start of code section
10
11 ;
12 ; t1.asm
13 ;
14 ; Copyright (C) 2018 dooleybl@tcd.ie
15 ;
16
17 ; function to calculate min(a, b, c)
18 ;
19 ; returns result in eax
20
21 public min                 ; make sure function name is exported
22
23 min: push ebp              ; push frame pointer
24      mov ebp, esp          ; update ebp
25
26      mov eax, [ebp+8]      ; v = a
27      mov ecx, [ebp+12]     ; ecx = b
28      cmp ecx, eax          ; if (b < v)
29      jge min_1             ;
30      mov eax, ecx          ; v = b
31
32 min_1: mov ecx, [ebp+16]    ; ecx = c
33      cmp ecx, eax          ; if (c < v)
34      jge min_2             ;
35      mov eax, ecx          ; v = c
36
37 min_2: mov esp, ebp         ; restore esp
38      pop ebp              ; restore ebp
39      ret 0                 ; return
40
```

### [t1.asm] - p

```
40
41 v ; function to calculate p(i, j, k, l)
42 ;
43 ; returns min(min(g, i, j), k, l) in eax
44
45 public    p                ; make sure function name is exported
46
47 v p:      push    ebp      ; push frame pointer
48           mov     ebp, esp  ; update ebp
49
50           push    [ebp+12]  ; push j onto stack
51           push    [ebp+8]   ; push i onto stack
52           push    g        ; push g onto stack
53
54           call    min       ; eax = min(g, i, j)
55
56           push    [ebp+20]  ; push l onto stack
57           push    [ebp+16]  ; push k onto stack
58           push    eax       ; push min(g, i, j) onto stack
59
60           call    min       ; eax = min(min(g,i,j), k, l)
61
62 v         mov     esp, ebp  ; restore esp
63           pop     ebp      ; restore ebp
64           ret     0        ; return
```

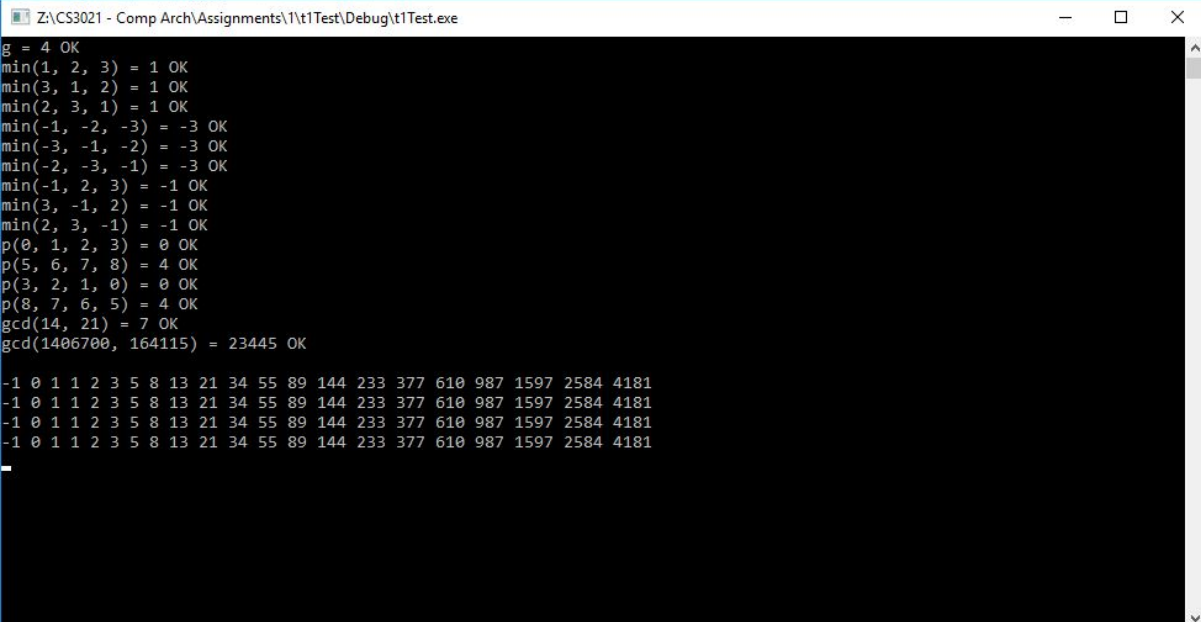
### [t1.asm] - gcd

```
66 v ; function to calculate gcd(a, b)
67 ;
68 ; returns gcd(a, b) in eax
69
70 public    gcd              ; make sure function name is exported
71
72 v gcd:     push    ebp      ; push frame pointer
73           mov     ebp, esp  ; update ebp
74
75           mov     eax, [ebp+12] ; eax = b
76           cmp     eax, 0     ; if(b==0)
77           je      gcd_retA   ; return a
78
79           mov     eax, [ebp+8] ; eax = a (dividend)
80           cdq      ; sign extend eax into edx
81           mov     ecx, [ebp+12] ; ecx = b (divisor)
82           idiv    ecx       ; edx = a % b
83
84           push    edx       ; push edx (a % b) onto stack
85           push    [ebp+12]  ; push b onto stack
86
87           call    gcd       ; eax = gcd(b, (a % b))
88           jmp     gcd_done  ;
89
90 gcd_retA:  mov     eax, [ebp+8] ; eax = a
91
92 v gcd_done: mov     esp, ebp  ; restore esp
93           pop     ebp      ; restore ebp
94           ret     0        ; return
```

[t1.h]

```
1  #pragma once
2
3  //
4  // t1.h
5  // Copyright(C) 2018 dooleybl@tcd.ie
6
7  // NB: "extern C" to avoid procedure name mangling by compiler
8  //
9
10 extern "C" int g;
11
12 extern "C" int _cdecl min(int, int, int);
13 extern "C" int _cdecl p(int, int, int);
14 extern "C" int _cdecl gcd(int, int);
15
```

### 3) Console Window



The screenshot shows a console window titled "Z:\CS3021 - Comp Arch\Assignments\1\t1Test\Debug\t1Test.exe". The output displays test results for three functions: min, p, and gcd. Each function is tested with multiple integer arguments, and the results are compared against expected values. The output is as follows:

```
g = 4 OK
min(1, 2, 3) = 1 OK
min(3, 1, 2) = 1 OK
min(2, 3, 1) = 1 OK
min(-1, -2, -3) = -3 OK
min(-3, -1, -2) = -3 OK
min(-2, -3, -1) = -3 OK
min(-1, 2, 3) = -1 OK
min(3, -1, 2) = -1 OK
min(2, 3, -1) = -1 OK
p(0, 1, 2, 3) = 0 OK
p(5, 6, 7, 8) = 4 OK
p(3, 2, 1, 0) = 0 OK
p(8, 7, 6, 5) = 4 OK
gcd(14, 21) = 7 OK
gcd(1406700, 164115) = 23445 OK
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
```