資芽 hand07 黃士育

1. (B)、(C)

2.

(a)

inline 可以直接將 function 內容展開，避免呼叫 function 的開銷（例如參數 push 到 stack memory）

(b)

遞迴每呼叫一次 function 就會產生新的 frame 造成額外的 stack memory，而迴圈僅需循環控制器跟變數，相比之下迴圈較遞迴節省資源所以效能更好一點，且若遞迴層級過深還會導致 stack overflow。

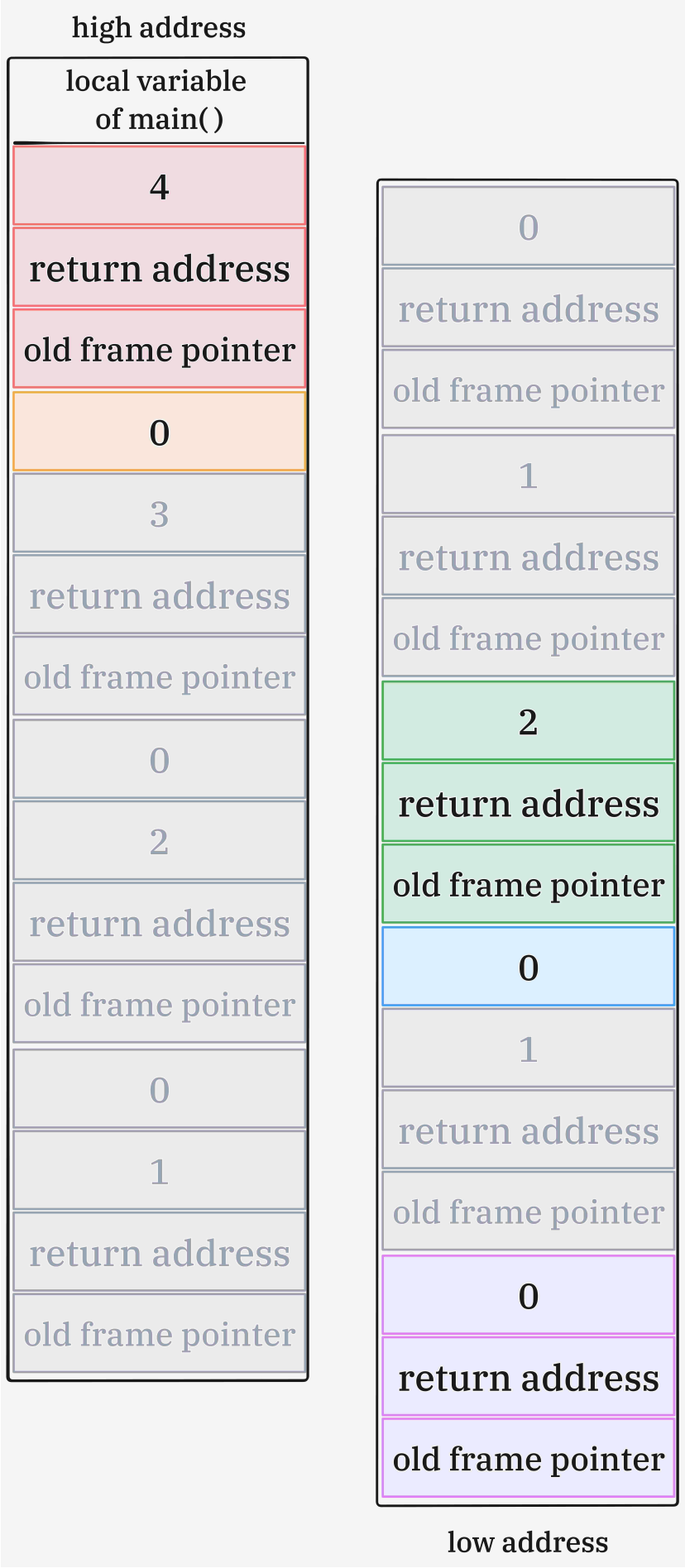
(c) 在背面

(d)

1. 在 main 裡面準備呼叫 func
   (a) 把參數 push 到 stack memory 上
   (b) 把 return 0 這條指令的位址 push 到 stack memory 上
   (c) 把程式的執行權交給 func。換句話說，就是把 program counter 設為 func 函數的第一條指令

2. 執行 func
   (a) 移動 stack pointer，在 stack memory 上劃分出一塊記憶體空間可以儲存 func 的所有區域變數。在這個例子的話是 12 bytes
   (b) 初始化 a, b, c 的值
   (c) 執行 printf （細節略）
   (d) 把回傳值 0 放到一個用來存放回傳值的系統變數裡
   (e) 把程式執行權交還給 main。換句話說，就是把 stack memory 上儲存的 return 0 位址給 pop 出來，把 pop 出來的這個值存給 program counter

3. 回去執行 main 剩下的東西
   (a) 把 stack memory 上殘留的 func 的參數 pop 掉
   (b) 執行 return 0（細節略）


high address
local variable of main( )
4
return address
old frame pointer
0
3
return address
old frame pointer
0
2
return address
old frame pointer
0
1
return address
old frame pointer
0
return address
old frame pointer
1
return address
old frame pointer
2
return address
old frame pointer
0
1
return address
old frame pointer
0
return address
old frame pointer
low address