|  |  |
| --- | --- |
| uint32\_t x, f(void) ;  x = f() ; | uint64\_t x, f(void) ;  x = f() ; |
| void f(int32\_t, int32\_t) ;  int32\_t a, b ;  f(a, b) ; | void f(int32\_t, int32\_t) ;  int64\_t a, b ;  f(a, b) ; |
| int32\_t f(int32\_t a)  {  return a + 1 ;  } | int64\_t f(int64\_t a)  {  return a + 1 ;  } |
| int32\_t f(int32\_t a)  {  int32\_t g(int32\_t) ;  return g(a + 1) ;  } | int32\_t f(int32\_t a)  {  int32\_t g(void) ;  return a + g() ;  } |
| int32\_t f(void)  {  int32\_t g1(void), g2(void) ;  return g1() + g2() ;  } | int32\_t product(int32\_t a, int32\_t b)  {  if (a < 0) return -product(-a, b) ;  if (a == 0) return 0 ;  return product(a – 1, b) + b ;  } |

|  |  |
| --- | --- |
| BL f  STR R0,x | BL f  STRD R0,R1,x |
| LDR R0,a  LDR R1,b  BL f | LDRD R0,R1,a  LDRD R2,R3,b  BL f |
| f: ADD R0,R0,#1  BX LR | f: ADDS R0,R0,#1  ADC R1,R1,#0  BX LR |
| f: PUSH {LR}  ADD R0,R0,#1  BL g  POP {LR}  BX LR | f: PUSH {LR,R4}  MOV R4,R0  BL g  ADD R0,R0,R4  POP {LR,R4}  BX LR |
| f: PUSH {LR,R4}  BL g1  MOV R4,R0  BL g2  ADD R0,R0,R4  POP {LR,R4}  BX LR | Product:  PUSH {LR,R4}  CMP R0,#0  BGE NotLT  NEG R0,R0  BL Product  NEG R0,R0  B Done  NotLT: BNE NotEQ  LDR R0,=0  B Done  NotEQ: SUB R0,R0,#1  MOV R4,R1  BL Product  ADD R0,R0,R4  Done: POP {LR,R4}  BX LR |