NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020 Final Exam (Part 2) Winter 2017

14. [5 pts ea] Convert each C function call into ARM assembly language.

C code	ARM Assembly

`int8_t *p8 ;`
`int32_t s32`
`int64_t a64[] ;`
`void f1(int8_t *, int32_t, int64_t []);`
 	
`f1(p8, s32, a64) ;`
	`LDR R0,p8`
	`LDR R1,s32`
	`ADR R2,a64`
	`BL f1`

`uint64_t *p64, *f2(void) ;`
 	
`p64 = f2() ;`
	`BL f2`
	`STR R0,p64`

`float average, reals[10] ;`
`void f3(float, float []) ;`
 	
`f3(average, reals) ;`
	`VLDR S0,average`
	`ADR R0,reals`
	`BL f3`

`float *presult ;`
`float f4(void) ;`
 	
`*presult = f4() ;`
	`BL f4`
	`LDR R0,presult`
	`VSTR S0,[R0]`

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020 Final Exam (Part 2) Winter 2017

15. [5 pts ea] Convert each C function definition into ARM assembly language.

C code	ARM Assembly

`int64_t f5(int32_t s32)`
`  {`
`  return s32 * s32 ;`
`  }`
	`f5:  MUL  R0,R0,R0`
	`  ASR  R1,R0,31`
	`  BX  LR`

`int32_t f6(float real)`
`  {`
`  return (int32_t) real ;`
`  }`
	`f6:  VCVT.S32.F32  S0,S0`
	`  VMOV  R0,S0`
	`  BX  LR`

`uint64_t f7(uint64_t u64)`
`  {`
`  return u64 + u64 ;`
`  }`
	`f7:  LSLS  R0,R0,1`
	`  ADC  R1,R1,R1`
	`  BX  LR`

`uint64_t f8(uint64_t u64)`
`  {`
`  return u64 >> 5 ;`
`  }`
	`f8:  LSR  R0,R0,5`
	`  ORR  R0,R0,R1,LSL 27`
	`  LSR  R1,R1,5`
	`  BX  LR`

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020 Final Exam (Part 2) Winter 2017

16. [5 pts ea] Convert each C function definition into ARM assembly language.

```
int32_t f9(int32_t x, int32_t a, int32_t b)
   {
   return (x < a || x > b) ;
   }
```

Do NOT use an IT block

```
f9:   CMP   R0,R1
      BLT   One
      CMP   R0,R2
      BLE   Zero
One:  MOV   R0,1
      BX    LR
Zero: MOV   R0,0
      BX    LR
```

```
int32_t f10(uint32_t score)
   {
   return (score >= 60 && score <= 100) ;
   }
```

Do NOT use an IT block

```
f10:  CMP   R0,60
      BLO   Zero
      CMP   R0,100
      BHI   Zero
One:  MOV   R0,1
      BX    LR
Zero: MOV   R0,0
      BX    LR
```

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

COEN 020 Final Exam (Part 2) Winter 2017

17. [5 pts ea] Convert each C function definition into ARM assembly language.

```
int32_t f11(uint64_t a64, uint64_t b64)
   {
   return (a64 <= b64) ;
   }
```

Use an IT block

```
f11:  SUBS  R0,R0,R2
      SBCS  R1,R1,R3
      ITE   LS
      MOVLS R0,1
      MOVHI R0,0
      BX    LR
```

```
int32_t f12(void)
   {
   int32_t f13(void), f14(void) ;

   return f13() + f14() ;
   }
```

```
f12:  PUSH  {R4,LR}
      BL    f13
      MOV   R4,R0
      BL    f14
      ADD   R0,R0,R4
      POP   {R4,PC}
```

```
int32_t f15(int32_t *p32, int32_t k)
   {
   return p32[k-1] ;
   }
```

```
f15:  SUB   R1,R1,1
      LDR   R0,[R0,R1,LSL 2]
      BX    LR
```

NAME: _____

SANTA CLARA UNIVERSITY
Department of Computer Engineering

18. [5 Pts] Write a function in ARM assembly language that calculates the single-length (32-bit) product of two 32-bit 2's complement signed operands. Unlike regular multiplication, the function should limit the product to no less than full-scale negative (0x80000000) and no more than full-scale positive (0x7FFFFFFF). The function prototype should be:

```
int32_t Product(int32_t a, int32_t b) ;

Product:   SMULL      R0,R1,R0,R1
           CMP        R1,R0,ASR 31
           BEQ        Done
           CMP        R1,0
           ITE        LT
           LDRLT      R0,=0x80000000
           LDRGE      R0,=0x7FFFFFFF
Done:      BX         LR
```

19. [5 pts] Write a C inline function that rotates its 32-bit unsigned argument left by 1 bit and returns the result. Use an extended asm statement inside the inline function so that the compiler is allowed to choose all of the registers.

```
static inline uint32_t RotateLeft1(uint32_t input)
{
uint32_t output ;

asm   (
      "ROR %[output],%[input],31       /t/n"
      :[output] "=r" (output)
      :[input] "r" (input)
      ) ;

return output ;
}
```