

Homework #7
Answers

>>>>>>>>>>Problem 12.31

```
!o                                ; optimal instruction set
; p1231.mas
; global variable
x:      dw      18
; =====
@f$ipi:  esba
        ldr 3          ; get q
        ldi          ; get *q
        addc 5         ; *q + 5
        push         ; place on the stack
        ldr 2          ; get p
        sti          ; *p = *q + 5
        reba
        ret
; =====
@g$i:    esba
        aloc 1         ; int y
        ldr 2          ; get x
        addc 10        ; x + 10
        str -1         ; y = x + 10
        ldc -1         ; get offset to y
        cora          ; &y
        push         ; parameter to f
        ldc 2          ; offset to x
        cora          ; &x
        push         ; parameter to f
        call @f$ipi
        dloc 2
        ldr 2          ; get x
        dout          ; display it
        ldc @msg0
        sout          ; display " "
        ldr -1         ; get y
        dout          ; display it
        ldc '\n'
        aout          ; display newline
        reba
        ret
@msg0:   dw      " "
; =====
main:    esba
        ldc 7
        push         ; parameter to g
        call @g$i
        dloc 1
        ldc 0         ; set up return value in ac
        ret
        public main
```

=====>>execution

```
% sim /z p1231
Simulator Version 5.0 Copyright (c) 2005 A. J. Dos Reis
Reading configuration file o.cfg
Reading microcode file o.hor
22 17
```

>>>>>>>>>>Problem 12.32

```
!o                                ; optimal instruction set
; p1232.mas
; =====
```

```

@f$ri:      esba
             ldc 45
             push                ; start the process toward using sti
             ldr 2                ; get pointer to array
             addc 3               ; get address of a[3]
             sti                  ; store mem[sp] into mem[ac]
             reba
             ret
; =====
@g$v:       esba
             aloc 10              ; int z[10]
             ldc 5
             push                ; int x = 5
             ldc -10             ; offset to the z array
             cora                 ; capture the address of z
             push                ; send parameter to f
             call @f$ri          ; call the function
             dloc 1              ; restore the stack
             ldc -10             ; offset to the z array
             cora                 ; capture address of z
             addc 3              ; get address z + 3
             ldi                  ; get the contents of (z + 3)
             dout                ; display as decimal
             ldc '\n'            ; << endl
             aout
             reba
             ret
; =====
main:       esba
             call @g$v           ; call function (no parameters)
             ldc 0                ; return 0
             reba
             ret
             public main

```

```

=====>>execution
avila% sim p1232 /z
Simulator Version 5.0 Copyright (c) 2005 A. J. Dos Reis
Reading configuration file o.cfg
Reading microcode file o.hor
45

```

>>>>>>>>>>Problem 13.34

```

!o                ; optimal instruction set
; p1334.mas
; =====
main:             esba
                 ldc 0                ; int i = 0
                 push
                 ldr 2                ; get argc
@L0:              jz @L1                ; exit if zero
                 subc 1               ; otherwise subtract 1
                 str 2                ; and store back in argc
                 ldr 3                ; get pointer to argv[0]
                 addr -1              ; get pointer to argv[i]
                 ldi                  ; get argv[i]
                 sout                 ; and display it (it's a string)
                 ldr -1               ; get i
                 addc 1               ; add 1
                 str -1               ; store back in i
                 ldc '\n'            ; << endl
                 aout
                 ja @L0               ; go around again
@L1:              ldc 0                ; return 0
                 reba

```

```
ret
public main
```

```
=====>>execution
```

```
avila% sim /z p1334 u v w
Simulator Version 5.0 Copyright (c) 2005 A. J. Dos Reis
Reading configuration file o.cfg
Reading microcode file o.hor
p1334
u
v
w
```

```
>>>>>>>>>Problem 13.36
```

```
!o
; p1336.mas
; =====
; global variable
a:          dw      0          ; int a;
; =====
@C@f$iri:   esba
            ldr 3          ; get the first parameter (a)
            push         ; place it on the stack
            ldr 2          ; get address of c (also c.x)
            sti          ; store a into c.x
            ldr 3          ; get a again
            push         ; place it on the stack
            ldc 1          ; offset into c to get y
            addr 2        ; add to &c to get &c.y
            sti          ; store a into c.y
            ldr 2          ; get address of c (also c.x)
            ldi          ; get value of c.x
            push         ; put it on the stack temporarily
            ldc 1          ; offset into c to get y
            addr 2        ; add to &c to get &c.y
            ldi          ; get value of c.y
            addr -1       ; compute c.x + c.y
            push         ; push x + y onto stack
            ldr 4          ; get &b parameter
            sti          ; b = x + y;
            reba
            ret
; =====
main:       esba
            aloc 2          ; C c;
            ldc a          ; get reference to a
            push         ; push parameter
            ldc 10         ; value of next parameter
            push         ; push next parameter
            ldc -2         ; offset to c
            cora          ; get address of c
            push         ; push implicit parameter
            call @C@f$iri ; c.f(10, a);
            dloc 3        ; restore stack
            ld a          ; get a
            dout         ; display it
            ldc '\n'      ; load newline
            aout         ; and display it
            ldc a          ; get reference to a
            push         ; push parameter
            ldc 20         ; value of next parameter
            push         ; push next parameter
            ldc -2         ; offset to c
            cora          ; get address of c
            push         ; push implicit parameter
```

```
call @C@f$iri ; c.f(20, a);
dloc 3 ; restore stack
ld a ; get a
dout ; display it
ldc '\n' ; load newline
aout ; and display it
ldc 30 ; get constant
str -2 ; store in c.x
ldr -2 ; get c.x
dout ; display it
ldc '\n' ; get newline character
aout ; and display it
ldc 0 ; return 0
reba
ret
```

```
; =====
public main
```

=====>>execution

```
% sim /z p1336
Simulator Version 5.0 Copyright (c) 2005 A. J. Dos Reis
Reading configuration file o.cfg
Reading microcode file o.hor
20
40
30
```