

1.

```
x:
.word 100
.LC0:
.ascii "x=%d\012\000"
main:
daddiu $sp,$sp,-48
sd $31,40($sp)
sd $fp,32($sp)
sd $28,24($sp)
move $fp,$sp
lui $28,%hi(%neg(%gp_rel(main)))
daddu $28,$28,$25
daddiu $28,$28,%lo(%neg(%gp_rel(main)))
sw $0,0($fp)
b .L2
nop

.L3:
ld $2,%got_page(.LC0)($28)
daddiu $4,$2,%got_ofst(.LC0)
ld $2,%call16(sprintf)($28)
move $25,$2
l: jalr $25
nop

lw $2,0($fp)
addiu $2,$2,1
sw $2,0($fp)
.L2:
lw $2,0($fp)
slt $2,$2,10
bne $2,$0,.L3
nop

nop
move $sp,$fp
ld $31,40($sp)
ld $fp,32($sp)
```

```
ld $28,24($sp)
daddiu $sp,$sp,48
j $31
nop
```

This instruction loops through and prints out values of x up to 10.

Output:

```
0
1
2
3
4
5
6
7
8
9
```

2.

```
.LC0:
.word 99
.word 81
.word -1
.word 76
.word 0
.word 1
.word 7
.word -74
.word 23
.word 99
.word -30
main:
daddiu $sp,$sp,-80
sd $fp,72($sp)
move $fp,$sp
lui $9,%hi(%neg(%gp_rel(main)))
daddu $9,$9,$25
daddiu $9,$9,%lo(%neg(%gp_rel(main)))
li $2,11 # 0xb
sw $2,12($fp)
ld $2,%got_page(.LC0)($9)
daddiu $3,$2,%got_ofst(.LC0)
ldl $4,%got_ofst(.LC0)($2)
ldr $4,7($3)
move $7,$4
daddiu $3,$2,%got_ofst(.LC0)
ldl $4,8($3)
ldr $4,15($3)
move $6,$4
daddiu $3,$2,%got_ofst(.LC0)
ldl $4,16($3)
ldr $4,23($3)
move $5,$4
daddiu $3,$2,%got_ofst(.LC0)
ldl $4,24($3)
ldr $4,31($3)
daddiu $3,$2,%got_ofst(.LC0)
ldl $8,32($3)
ldr $8,39($3)
move $3,$8
sd $7,16($fp)
sd $6,24($fp)
sd $5,32($fp)
```

```
sd $4,40($fp)
sd $3,48($fp)
daddiu $2,$2,%got_ofst(.LC0)
lw $2,40($2)
sw $2,56($fp)
sw $0,4($fp)
sw $0,8($fp)
sw $0,0($fp)
b .L2
nop
```

```
.L5:
lw $2,0($fp)
dsll $2,$2,2
daddu $2,$fp,$2
lw $2,16($2)
lw $3,4($fp)
slt $2,$3,$2
beq $2,$0,.L3
nop
```

```
lw $2,0($fp)
dsll $2,$2,2
daddu $2,$fp,$2
lw $2,16($2)
sw $2,4($fp)
.L3:
lw $2,0($fp)
dsll $2,$2,2
daddu $2,$fp,$2
lw $2,16($2)
lw $3,8($fp)
slt $2,$2,$3
beq $2,$0,.L4
nop
```

```
lw $2,0($fp)
dsll $2,$2,2
daddu $2,$fp,$2
lw $2,16($2)
sw $2,8($fp)
.L4:
lw $2,0($fp)
addiu $2,$2,1
sw $2,0($fp)
.L2:
```

```
lw $3,0($fp)
lw $2,12($fp)
slt $2,$3,$2
bne $2,$0,.L5
nop

move $2,$0
move $sp,$fp
ld $fp,72($sp)
daddiu $sp,$sp,80
j $31
nop
```

This program goes through the array arr of size 11 in this example, and organizes the array based on the highest and lowest values of the array. At the end, the array should be organized. I was unable to get this to compile.

3.

main:

daddiu \$sp,\$sp,-832

sd \$fp,824(\$sp)

move \$fp,\$sp

sw \$0,0(\$fp)

b .L2

nop

.L3:

lw \$3,0(\$fp)

lw \$2,4(\$fp)

mult \$3,\$2

mflo \$3

lw \$4,4(\$fp)

lw \$2,0(\$fp)

dsll \$2,\$2,1

dsll \$5,\$2,2

daddu \$2,\$2,\$5

daddu \$2,\$2,\$4

dsll \$2,\$2,2

daddu \$2,\$fp,\$2

sw \$3,16(\$2)

lw \$2,0(\$fp)

addiu \$2,\$2,1

sw \$2,0(\$fp)

.L2:

lw \$3,0(\$fp)

lw \$2,8(\$fp)

slt \$2,\$3,\$2

bne \$2,\$0,.L3

nop

sw \$0,0(\$fp)

b .L4

nop

.L5:

lw \$3,4(\$fp)

lw \$2,0(\$fp)

dsll \$2,\$2,1

dsll \$4,\$2,2

daddu \$2,\$2,\$4

daddu \$2,\$2,\$3

dsll \$2,\$2,2

```

daddu $2,$fp,$2
lw $3,16($2)
lw $4,0($fp)
lw $2,4($fp)
dsll $2,$2,1
dsll $5,$2,2
daddu $2,$2,$5
daddu $2,$2,$4
dsll $2,$2,2
daddu $2,$fp,$2
sw $3,416($2)
lw $2,4($fp)
addiu $2,$2,1
sw $2,4($fp)
.L4:
lw $3,0($fp)
lw $2,8($fp)
slt $2,$3,$2
bne $2,$0,.L5
nop

move $2,$0
move $sp,$fp
ld $fp,824($sp)
daddiu $sp,$sp,832
j $31
nop

```

This program first populates the array a with elements, these elements are generated by $a[i][j] = i*j$.

After the array is populated another loop is run to transpose the elements within array a. These are placed into array transpose. This transpose operation will produce a new matrix by interchanging the rows and columns of the original array a. I was unable to produce proper compilation.