Benjamin Smith
Email: bxs566@case.edu
Course: CSDS 337 - Compiler Design
Instructor: Dr. Vipin Chaudhary

Problem Set - 6
ID: 3559750
Term: Spring 2024
Due Date: 24^{th} April, 2024

Number of hours delay for this Problem Set: 0.
Cumulative number of hours delay so far: 56.

I discussed this homework with: No one.

Problem 1 - 10 points
Generate code for the following three-address statements assuming `a` and `b` are arrays whose elements are 4-byte values. The four-statement sequence

```
x = a[i]
y = b[j]
a[i] = y
b[j] = x
```

Solution:

Problem 2 - 10 points
Determine the cost of executing the following.

1.
```
LD R0, c
LD R1, i
MUL R1, R1, 8
ST a(R1), R0
```

2.
```
LD R0, p
LD R1, 0(R0)
ST x, R1
```

Solution:

Problem 3 - 10 points
Below is code to count the number of primes from 2 to n , using the sieve method on a suitably large array a. That is, a [i] is TRUE at the end only if there is no prime $\sqrt{i}$ or less that evenly divides i. We initialize all a[i] to TRUE and then set a[j] to FALSE if we find a divisor of j.

```
for (i=2; i<=n; i++)
    a[i] = TRUE;
count = 0;
s = sqrt(n);
for (i=2; i<=s; i++)
    if (a[i]) /* i has been found to be a prime */ {}
        count++;
        for (j=2*i; j<=n; j = j+i)
```

```
            a[j] = FALSE; /* no multiple of i is a prime */
    }
```

a Translate the program into three-address statements of the type we have been using in this section. Assume integers require 4 bytes.

b Construct the flow graph for your code from (a)

c Identify the loops in your flow graph from (b).

Solution:

Problem 4 - 10 points
Suppose a basic block is formed from the C assignment statements

```
x = a + b + c + d + e + f;
y = a + c + e;
```

a Give the three-address statements (only one addition per statement) for this block.

b Use the associative and commutative laws to modify the block to use the fewest possible number of instructions, assuming both x and y are live on exit from the block.

Solution:

Problem 5 - 20 points
Consider the expression $(a - b) + e * (c + d)$.

a Generate optimized code using three registers.

b Generate optimized code using two registers.

Solution:
