Answer the following questions in the space provided. The exam has 150 points total, 25 points for each of the six questions. No notes, no book, and no neighbors.

1. Hacker Hank is writing a Java program, and decided that in addition to the “while (expression) statement” form of loop, Java should also have a statement of the form “statement until(expression)”. Ignoring for the moment that java already HAS a post-test loop statement, Hank’s friend Ada is able to write a single statement using this type of loop to demonstrate to hank that his proposal is a very BAD idea. What sort of statement did Ada write and why is hank’s proposal a bad idea?

2. Explain why register-style assembly language code has faster execution than stack-style assembly language. Give at least 3 reasons why generating register-style code is more difficult than generating stack-style assembly language.

3. Explain the optimization termed reduction in strength. What is the purpose of this optimization? What are the steps involved in performing this optimization? Give an intermediate representation (e.g. CFG) for a short example program and illustrate how reduction in strength would change the program.

4. Explain the relationship between symbol tables and activation records.

5. Suppose we added the boolean data type to our language. Go through each of the different phases of the compiler, and explain what changes we would have to make to include this feature. The following example program illustrates the use of this data type

```plaintext
function foo (x : int, y : int) 
var flag : boolean;
begin 
    flag := x < y;
```

function foo (x : int, y : int) 
var flag : boolean;
begin 
    flag := x < y;
while (flag)
  flag := not flag;

6. Describe in pseudo-code the assembly language that would be generated for the following bit of code:

function exam ( x : int)
  var a : [2 : 17] int;
begin
  if (((x > 5) and (x < 17)) or (x == 3)) printint(a[x] + 1)
end

any reasonable approximation to assembly language accepted