NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING
EXAMINATION FOR
Semester 1, 2011/2012

CS4212 — Compiler Design

November 2011 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. The examination paper contains **FIVE (5) questions** and comprises **FIFTEEN (15) pages.**
2. The maximum attainable score is 50.
3. All questions must be attempted for the maximum score to be attained.
4. This is an OPEN BOOK exam.
5. Write all your answers in the space provided in this booklet.
6. **Please write your matriculation number below.**

MATRICULATION NUMBER: _____

(this portion is for the examiner's use only)

Question	Marks	Remark
Q1		
Q2		
Q3		
Q4		
Q5		
Total		

Question 1 [8 marks]

Lexical Analysis

A. For each of the following, explain why you're not surprised that there is no regular expression defining it:

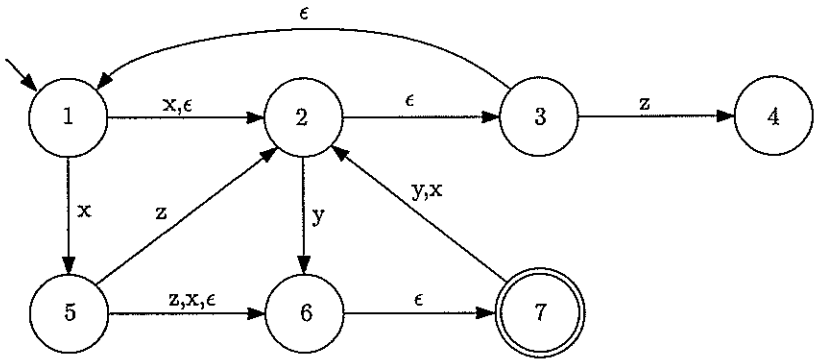
(a) Strings of a's and b's where there are more a's than b's.

(b) Strings of a's and b's that are palindromes.

(c) Syntactically correct Java programs.

[3 marks]

B. Convert the following NFA to a DFA. Specify the DFA states as sets of NFA states, so that your application of the conversion algorithm is clear.



[5 marks]

Question 2 [10 marks]

Syntactic Analysis

A. The language of regular expressions is itself a context-free language. Devise a non-ambiguous grammar that generates this language. (Implement only the |, sequencing, and * operators). [4 marks]

B. Based on the model given in Lecture 9, write (in Prolog) a syntax analyzer for the language specified by your grammar. Make the syntax analyzer generate a syntax tree for the regular expression given as input. [6 marks]

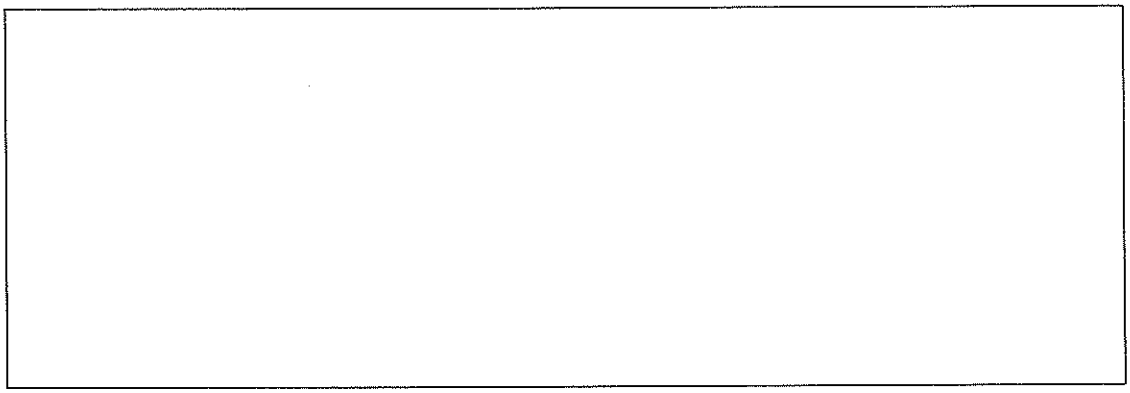
Question 3 [10 marks]

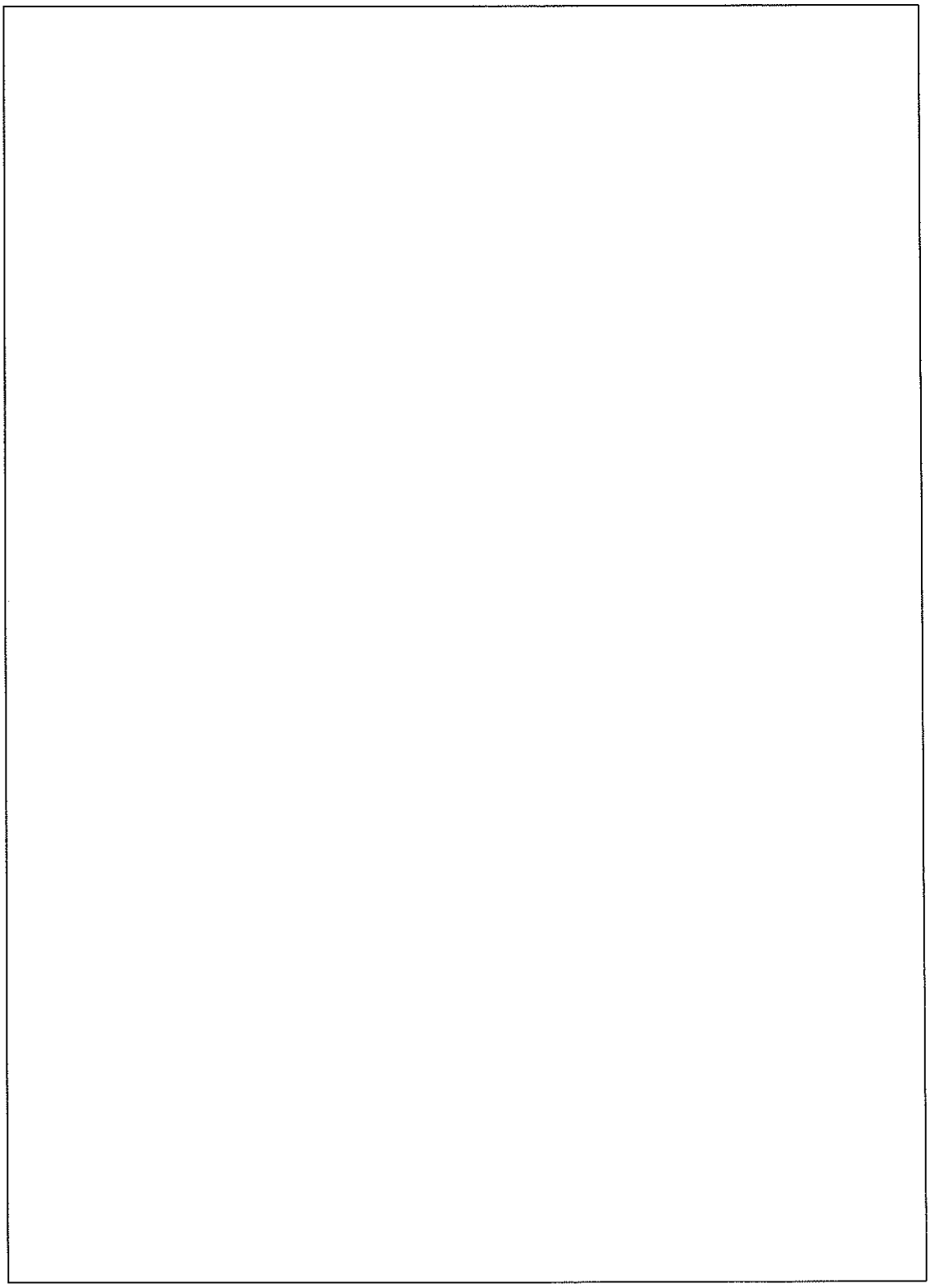
The language Python has a statement called `yield`, that can be used inside a procedure, as a replacement for the `return` statement. We shall explain how the `yield` statement works by means of an example. The text between the character # and the end of line represents comments.

```
>>> def gen():
    n = 1
    while True:
        yield n # returns n; saves current execution point
        n += 1  # continuation point for next(handle)

>>> handle = gen() # creates a procedure handle
>>> next(handle) # every next saves the state of procedure
1
>>> next(handle) # execution is resumed from right after last 'yield'
2
>>> next(handle) # can be applied an unbounded no of times
3
>>> next(handle)
4
```

The procedure `gen` uses the statement `yield`, instead of return. Such a procedure is called a *generator*, and, when called, will return a *handle*. Subsequently, the predefined procedure `next` can be applied to the handle to execute the code inside the procedure. This `yield` statement will return its argument to the outside environment. However, it will also save the execution state of the procedure, such that subsequent calls of `next(handle)` will resume the execution from right after the most recently executed `yield`. The local variables will also be preserved, so that the call `next(handle)` appears to *continue* the execution of the code. There is no limit on the number of `next` applications.

Devise a compilation scheme for this kind of procedure. Explain in as much detail as possible how you would envisage the handling of the activation record. Pay attention to the fact that multiple handles may be active for a generator simultaneously, each with its own state, in a fashion similar to the way multiple objects of the same class may be active at the same time.





Question 4 [10 marks]

The compilation scheme presented in class for object oriented code has the disadvantage that each object is translated into a structure that contains pointers to all the methods of the object's class. This is in general wasteful, since these pointers are (at least for languages such as Java and C++) the same, and so they are repeated in the storage of each object.

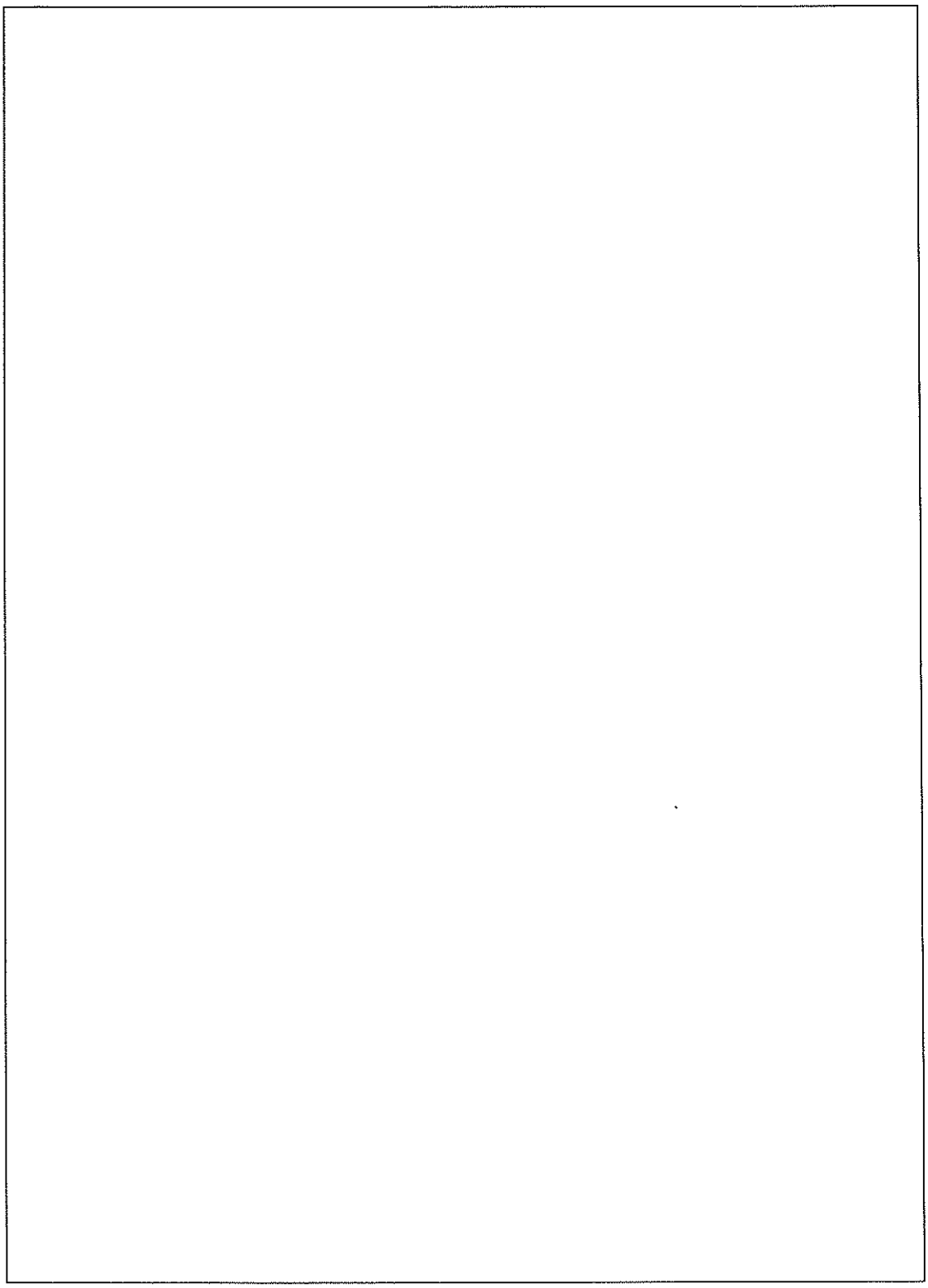
Devise a more efficient compilation scheme, where each class has a unique method pointer record, and each object has a reference to this record.

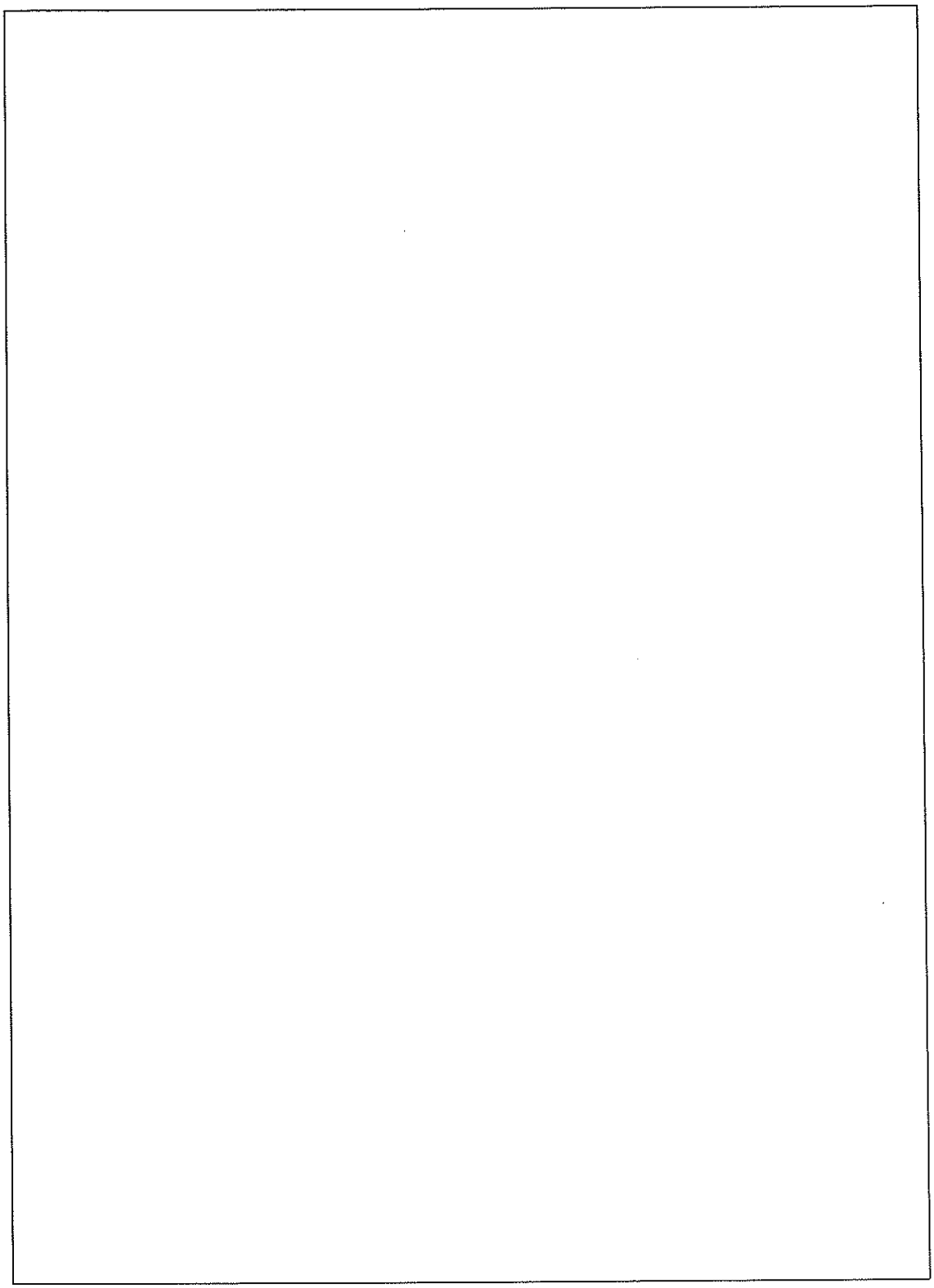
A. Describe your compilation scheme in as much detail as possible. Pay special attention to describing what new data structures you would add, what new procedures, and explain clearly when these new procedures would be called. [5 marks]

B. Use your translation scheme to translate the following Java program into C.

```
class GeomObj {
    public Boolean isDrawable() { return false ; }
}

class Drawable {
    public Boolean isDrawable() { return true ; }

    public void draw() {
        System.out.println("Generic drawable object") ;
}

    public static void main(String [] argv) {
        GeomObj x ;
        if ( argv[0].equals("s") ) {
            x = new Square(10,10,10) ;
        } else {
            x = new Rectangle(5,5,5,20) ;
        }
        if ( x.isDrawable() ) x.draw() ;
    }
}

class Rectangle extends Drawable {
    int x, y, sidex, sidey ;
    Rectangle(int x, int y, int sidex, int sidey) {
        this.x = x ;
        this.y = y ;
        this.sidex = sidex ;
        this.sidey = sidey ;
    }
    public void draw() {
        System.out.println("Rectangle with corner at ("
                           +x+","+y+") and "
                           +"sides "+sidex+" and "+sidey) ;
    }
}

class Square extends Rectangle {
    Square(int x, int y, int side) {
        super(x,y,side,side) ;
    }
    public void draw() {
        super.draw() ;
        System.out.println("It's actually a square!") ;
    }
}
```

[5 marks]





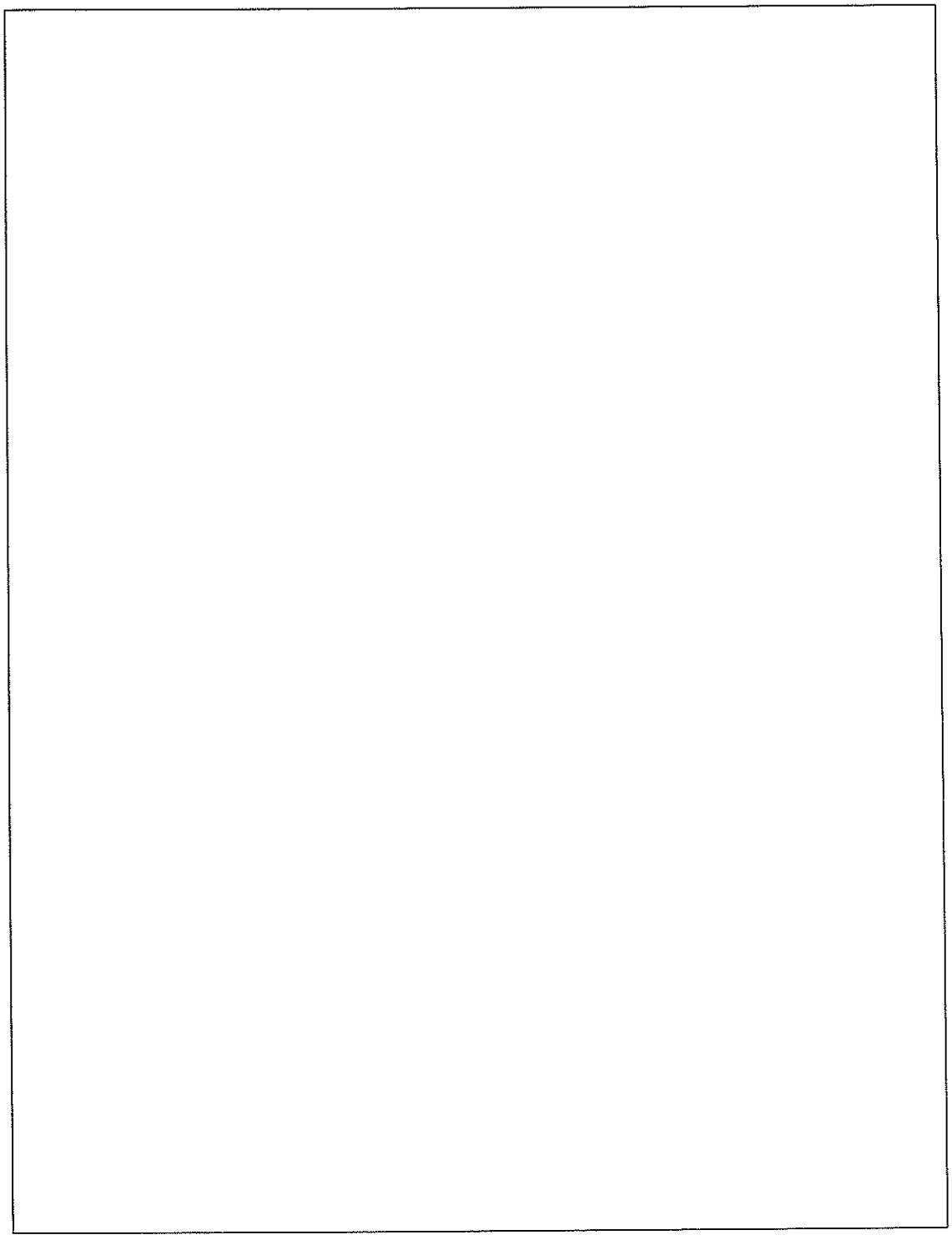
Question 5 [12 marks]

Consider the following TAC code.

```
          m=0
          v=0
l1::      if v>=n goto l5
          r=v
          s=0
l2::      if r<n goto l3
          v = v+1
          goto l1
l3::      x = [r]
          s = s+x
          if s<=m goto l4
          m = s
l4::      r = r+1
          goto l2
l5::
```

A. Break this program into basic blocks and draw its control flow graph. [3 marks]

B. Perform liveness analysis on this code snippet. Calculate the live-in and live-out at each statement. [3 marks]



C. Perform reaching definitions analysis on the code snippet given above. [3 marks]

D. Explain why the above analyses may be useful in the workings of the compiler. [3 marks]

— END OF PAPER —