

Midterm Examination
May 11, 2005

Name (Last, First): _____, _____ Score: _____/100

Directions

- This exam consists of 5 questions on 7 pages; be sure that you have the entire exam before starting.
- Make sure that you write your name on every page of the exam.
- Answer each question in the space following the question. If you find it necessary to continue an answer elsewhere, indicate clearly the location of its continuation and label its continuation with the question number.
- The point value of each question is indicated at its beginning. The entire exam has 100 points.
- Be concise and indicate clearly what your answer is. Presentation and simplicity of your answers may affect your grade.
- This exam is open book open notes.
- You have **50** minutes.

Problem 1 (20 Points)

Problem 1

This problem tests your familiarity with BNF, EBNF, parse trees etc.

1.a (6 Points) Convert the following BNF into EBNF:

$$\begin{aligned} \langle E \rangle & ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle \\ \langle T \rangle & ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle \\ \langle F \rangle & ::= \text{id} \end{aligned}$$

In this grammar, $\langle E \rangle$, $\langle T \rangle$, and $\langle F \rangle$ are nonterminals, and '+', '*', and id are terminals. Also assume that $\langle E \rangle$ is the starting nonterminal.

1.b (4 Points) Draw the syntax diagram for the EBNF grammar.

Problems 1.c and 1.d are based on the following BNF grammar, G_1 :

Problem 1

$$\begin{aligned} \langle UT \rangle & ::= \langle UT \rangle ? \langle UV \rangle \mid \langle UV \rangle \\ \langle UV \rangle & ::= \langle UW \rangle ! \langle UV \rangle \mid \langle UW \rangle \\ \langle UW \rangle & ::= \text{term} \end{aligned}$$

In this grammar, $\langle UT \rangle$, $\langle UV \rangle$, and $\langle UW \rangle$ are nonterminals, and $?$, $!$ and *term* are terminals. Also assume that $\langle UT \rangle$ is the starting nonterminal.

1.c (6 Points) Answer the following for the operators $?$ and $!$:

- Which of the operators are left-associative?
- Which of the operators are right-associative?
- List the operators in the order of precedence. If two operators have the same precedence, indicate so.

1.d (4 Points) Show the parse tree (if any) for the following strings: $i ! i ? i ? i ! i$

Problem 2 (20 Points)

Problem 2

The following BNF grammar defines a language L :

$$\begin{aligned} \langle S \rangle & ::= \langle T \rangle \langle U \rangle \\ \langle T \rangle & ::= a \langle T \rangle b \mid \epsilon \\ \langle U \rangle & ::= c \langle U \rangle d \mid \epsilon \end{aligned}$$

In this grammar $\langle S \rangle$, $\langle T \rangle$, and $\langle U \rangle$ are nonterminals, whereas a , b , c , and d are terminals. $\langle S \rangle$ is the starting nonterminal.

2.a (3 Points) Compute the first sets of $\langle S \rangle$, $\langle T \rangle$, and $\langle U \rangle$.

2.b (17 Points) Construct a predictive parser for the language. Extend your parser such that it outputs the number of a 's, b 's, c 's, and d 's in an input string. You do not have to write the parser part separately from the part that performs counting. You can write one program that does both parsing and counting. Also, feel free to write the program in pseudo-code. However, make sure that you explicitly state any assumptions that you make.

Problem 3 (20 Points)

Problem 3

This problem deals with a special two-dimensional $N \times N$ array. Assume that the row and column indices start at 1. Also, assume that N is even. As shown the figure 1(a), the two matrices of size $(N/2) \times (N/2)$ contain zero values.

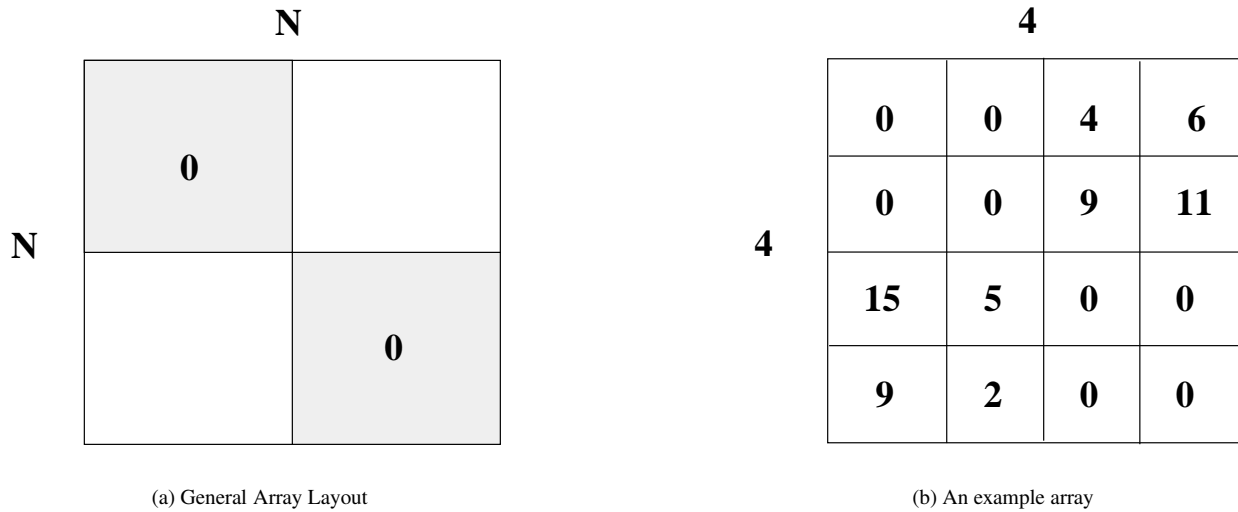


Figure 1: Problem 3 Matrix

In Figure 1(b) we show an example of such a 4×4 matrix. For this question, assume that the elements of arrays are laid out in the storage in the row-major form. **Note that zero elements are not stored in the memory.**

3.a) (4 Points) Show the memory layout of the elements of the array in Figure 1(b).

3.b) (16 Points) Give a general formula for the address of element $A[i][j]$ where A is an $N \times N$ array.

Problem 4 (25 Points)

Problem 4

The program below shows both the class definitions and parts of the main program that uses the class definitions. The output of running `test` is shown in the table. You will need to do the following: (i) Find any type errors. If any, replace it by reassigning variables. You can make any valid variable assignments as long as it produces the right output. (ii) Use the output to complete `test`. In other words, fill in the statements that will cause the corresponding output to occur. In some cases, there are more than one solutions possible. You only need to show one.

Class Definition

```
class A {
    A val;
    public void Print1(A a) {
        val = a;
        Print2();
        val.Print2();
        System.out.println();
    }
    public void Print2() {
        System.out.print("A ");
    }
}
class B extends A {
    public void Print2() {
        System.out.print("B ");
    }
}
class C extends B {
    public void Print2() {
        System.out.print("C ");
    }
}
class D extends C {
    public void Print2() {
        System.out.print("D ");
    }
}
```

Test class

```
class test {
    public static void main(String argv[])
    {
        A a1;   B b1; C c1; D d1;
        a1 = new A();           b1 = new B();
        c1 = new C();           d1 = new D();

        ??????????             // stmt 1

        d1 = a1;

        ??????????             // stmt 2

        b1 = a1;
        b1 = d1;

        ??????????             // stmt 3

        a1 = new B();

        c1 = b1;

        c1 = d1;
        ??????????             // stmt 4

        d1 = a1;

        b1 = c1;

        ??????????             // stmt 5
    }
}
```

Statement	Output
1	A D
2	B C
3	C B
4	B C
5	D D

Problem 5 (15 Points)

Problem 5

This problem deals with static and dynamic scoping. Assume a language, L , that combines both static and dynamic scoping: all variables (global, parameters and local) that start with the prefix `''static_''` follow static scope; others starting with the prefix `''dynamic_''` follow dynamic scoping. Show the output of the following program.

```
int static_x;                // global statically scoped variable
int dynamic_y;              // global dynamically scoped variable

void f(int static_x) {
    dynamic_y = dynamic_y - 1;
    printf (``%d %d\n``, static_x, dynamic_y);
    if (static_x <= 0) return;
    else
        g(static_x-1);
    printf (``%d %d\n``, static_x, dynamic_y);
}

void g(int dynamic_y) {
    static_x = static_x -1;
    printf (``%d %d\n``, static_x, dynamic_y);
    if (dynamic_y <= 0) return;
    else
        f(dynamic_y-1);
    printf (``%d %d\n``, static_x, dynamic_y);
}

void main() {
    static_x := 3; dynamic_y := 2;
    f(static_x);
    printf (``%d %d\n``, static_x, dynamic_y);
}
```

Output =