

CS328, Fall 1999, Test 1

Time 75 min. Use extra paper as needed, but make sure to identify each answer.

YOU MUST RETURN THIS PAGE. NAME _____

- 1 You have a two-stage Pascal compiler on machine M. This compiler is a compiler into P-code, for which P-code you have an abstract machine in the form of a software interpreter. You write a program P-Pascal (program P in Pascal).
 - a) Using tombstone notation, show what must be done to execute the program on that machine.
 - b) Show exactly, using tombstone notation, what must be done every time to repeat that execution.
- 2 You decide the execution of P-Pascal is too slow (you have to run the program from time to time). How would you speed that execution? Considering cost, come up with a suggestion and show how this would allow faster execution of your P-Pascal.
- 3 Assume that opcodes are numbered starting at opcode 1, using dictionary ordering of the corresponding mnemonic names. Instructions are: 1 byte for opcode, and 1 byte for each argument. Arguments in machine code are addresses - offsets in a relocatable segment, with the segment identified by a base register (that is, actual addresses are computed on the fly by combining base register and the machine-code offsets). Variables are 4 bytes (Space directive), aligned to multiple-of-four. Literal pool is placed after the code and its variables, following the same principles (size and alignment). There is only 1 segment for all code, variables, and literals.
 - a) Show the load module (relocatable machine code) for the following program (in an accumulator architecture)
Load 10
Mult 2
Store X
Copy X Y
Stop
X Space
Y Space
 - b) What constraints we have on data and program?
- 4 Develop a BNF grammar for the following expressions:
unary operators & (address) * (dereference)
binary operators + * , (comma)
ternary ? :
The above are a subset of C operators, with their meaning, associativity, and precedence preserved. Do not allow (), and assume only numbers are allowed.
- 5 You are asked to design a deterministic FA (graphically), for recognizing the following three tokens. Assume tokens are separated by WS, and that alphabet is all letters and digits
valid tokens start with letters, end with digits, and no letter can follow a digit

there must be at least one letter and one digit

token1 is when there are more letters than digits, token2 is when the numbers are equal, token3 otherwise. For example,

token1 examples: aaa2, bcz1

token2 examples: aa22, a1

token3 examples: a223

a) Can it be done if the total length can be at most 4 (otherwise it is an error)? If not, why not? If yes, draw the FA.

b) Can it be done if the total length is up to 32? Why yes or no?

c) Can it be done if we have no limits? Why yes or no?

Make sure to clearly identify states recognizing specific tokens. by placing t1, t2, or t3 inside of those states. Do not use error states.