**QUIZ 1** **AMAT 2120** **October 24, 2004**

NAME ANSWERS ID

a. The term *byte* means

□ certain type of C variables

□ certain type of variables in a programming language other than C

☑ 8 bits

□ 256 bits

□ the range of integers from −128 to 127.

(The first two answers are badly wrong. The last two are incorrect, although numbers are relevant. A byte is a sequence of eight binary digits (0 or 1) called bits. There are $2^8 = 256$ different sequences, or arrangements, of length 8 that can be represented by a byte. They can, for instance, represent nonnegative numbers from 0 to 255 or, alternatively, signed numbers from −128 to +127.)

b. If the variable **n** is **int** and the variable **x** is float, then the assignment **n=x;** is

□ illegal; it will cause compilation error

□ safe and sound

☑ will pass compilation but may lead to a mathematical error due to roundoff effect

□ will cause no problems for the program, but isn't best in terms of style; better use the explicit cast: **n=(int)x;**

(If the programmer really wants to discard the fractional part of $x$, the last answer gives just fine suggestion. Otherwise the checked answer is correct.)

c. Assuming that $x$ is a variable of type `int`, the value of **0&&x** is

☑ always 0

□ 0 or 1 depending on the value of $x$

□ 0 or $x$ depending on the value of $x$

□ unpredictable

(The last answer is badly wrong and the next-to-last one is plain wrong. The 2nd answer is reasonable and it would be formally correct if it said "*possibly* depending".

d. The statement `#define _DEBUG` is

☑ preprocessor directive

□ definition of function

□ declaration of the variable `_DEBUG`

□ instruction to the compiler to process the code in debugging mode

(Everything that begins with `#` character (and not a part of comments) is a preprocessor directive. Other examples are `#include<..>`, `#ifdef` … . The word `_DEBUG` is the name of a symbolic constant but not of a variable. The value of that constant is an empty string, but one is free to "`#define _DEBUG 3.1415`", in which case the value of the symbolic constant `_DEBUG` would be the string "3.1415". The last answer may have some truth in it, but as stated it is technically incorrect. We can elaborate to reveal the truth in a correct form. Namely, the code may contain some parts that should only be compiled in the debugging mode and some parts that should be omitted in debugging mode. If the former are placed between the `#ifdef _DEBUG` and the subsequent `#else` and the latter are between that `#else` and subsequent `#endif`, then the line `#define _DEBUG` in program's preamble indeed causes the compiler to process only the debugging mode relevant code.)

e. The command `cd` without arguments

□ changes current directory to the root directory

☑ changes current directory to user's home directory

□ creates an unnamed directory

□ creates root directory

(The first answer describes the effect of command "`cd /`". The abbreviation `cd` is indeed confusing; the name of command to create directory is `mkdir`. Yet the last two answers are worse than just command name mix-up. There is no such thing as unnamed directory. And no one, sysadmin including, ever creates root directory. It exists since the very moment a Unix-type operating system (e.g. Linux) is installed on the computer. One more remark: regular users cannot create their home directories; this is done for them by sysadmin.)