

COS 217 Midterm

Fall 2007

Please write your answers clearly in the space provided. For partial credit, show all work. State all assumptions. You have exactly 50 minutes for this exam. This midterm is open book, open notes. Put your name on every page. Write out and sign the Honor Code pledge just before turning in the test. *“I pledge my honor that I have not violated the Honor Code during this examination.”*

Question	Score
1	/20
2	/40
3	/40
Total	/100

Name:

Honor Code:

1 Short Answer

Answer the following in only the space provided.

1. What is the base 10 value of this expression: $((0xFA / 02) \gg 1) \& 017 - 07$
2. What is the base 10 value of the following twos-complement 4-bit number: 0101
3. What is the base 10 value of the following twos-complement 1-bit number: 1
4. What's abstract about an ADT?
5. Why does the stack usually grow toward the heap?
6. Express this type cast in English: $(void (*) (void *, int, char*))$
7. In C, write the worst hash function you can. In what ways is it evil (must have at least one)?

2 DFA

Draw a DFA that recognizes ordinary floating point numbers (examples include: “0.1”, “.5”, “4.1”, “9.”, and “0.0”, but do not include integers such as “47”, scientific, and other notations). The tokens in your input consist of LETTERS, NUMERALS, SPACES, and DOTs. A floating point number does not include a SPACE, but a SPACE must precede it. Each transition in your DFA should be labelled for the input token(s) consumed. Give each state a unique descriptive name. Identify your start state. In addition, label each state as either FLOAT or NOT. FLOAT states indicate that the current position is inside a valid floating point number. The NOT states indicate otherwise. For example, the string: “Hi 4.67X” would have the following actions/states (where \$ indicates the current position):

```
$          NOT
(transition which consumed a LETTER)
H$        NOT
(transition which consumed a LETTER)
Hi$       NOT
(transition which consumed a SPACE)
Hi $      NOT
(transition which consumed a NUMERAL)
Hi 4$     NOT (looks like an integer)
(transition which consumed a DOT)
Hi 4.$    FLOAT
(transition which consumed a NUMERAL)
Hi 4.6$   FLOAT
(transition which consumed a NUMERAL)
Hi 4.67$  FLOAT
(transition which consumed a LETTER)
Hi 4.67X$ NOT
```

3 Bug Hunt!

The following code on the next page tries to implement a palindrome checker. The current status of the program is that it compiles, albeit with several warnings. So don't worry about syntax errors. Answer the following questions.

1. There are flaws (mostly logical) on the following lines: 43, 44, 45, 46, 19, 24, 28, 30, 51-56. Fix the flaws on the code sheet itself. Also fix a programming style issue on lines 10, 11. Make minimal changes. The man page description of the function `gets` reads:
`gets()` reads a line from `stdin` into the buffer pointed to by `s` until either a terminating newline or EOF, which it replaces with the null termination character.
2. Enhance the program so that it can handle string inputs of arbitrary length. Write a C code snippet showing only the changes, in the space below. You do not have to rewrite the entire program.

```

0  /*****
1  Name: palindrome
2  Description: Prompts the user to enter a
3  string upto 15 characters long and
4  checks to see if it is a palindrome.
5  *****/
6
7  #include <stdio.h>
8  #include <string.h>
9
10 #define TRUE 1
11 #define FALSE 0
12
13 int checkPalindrome(char *pcForward, char *pcReverse) {
14     int ispalindrome = TRUE;
15
16     /* Check if the first and last characters are equal,
17      * then move inward checking each pair of characters
18      */
19     while(pcForward <= pcReverse && ispalindrome);
20     {
21         /* If corresponding characters do not match
22          * then the word is not a palindrome
23          */
24         if(pcForward != pcReverse) {
25             ispalindrome = FALSE;
26         } /* end if */
27         pcForward++;
28         pcReverse++;
29     } /* end while */
30     return &ispalindrome;
31 } /*end checkPalindrome*/
32
33 int main () {
34     /* Create a character array */
35     char string[15];
36     int ispalindrome; /* boolean */
37     char *pcForward,*pcReverse; /*pointers to traverse the
38                                 string from the front and
39                                 the back in the forward and
40                                 reverse directions respectively*/
41
42     /* prompt the user to input the string */
43     printf("Enter a string: ");
44     gets(string);
45     pcReverse = strlen(string);
46     pcForward = 0;
47
48     ispalindrome = checkPalindrome(pcForward, pcReverse);
49
50     /* print out the results */
51     switch (ispalindrome) {
52         case TRUE:
53             printf("%s is a palindrome!\n", string);
54         case FALSE:
55             printf("Sorry, %s is not a palindrome\n", string);
56     }
57     return 0;
58 } /* end main */

```