CSI 333 – Programming at the Hardware-Software Interface – Fall 2012

First Midterm Examination – October 9, 2012

Instructions: This examination has five questions for a total of 100 points. Answer all questions. Write all your answers in your blue book.

Question I (12 points total)

(a) When the function scanf is used, we must pass a pointer to the variables whose values are to be read in. Why? (2 points)

(b) Give one reason why a call to the function fclose may fail. (2 points)

(c) Write a C function count_nine with the following header:

\[
\text{int count_nine (char s[])}
\]

The input parameter \text{s} is a string that is correctly terminated by '\0'. The function should return the number of times the character '9' occurs in the string given by the parameter \text{s}.

You need to show the C code only for the function. There is no need to include comments and you may use magic numbers in your code. (8 points)

Question II (33 points total)

Note: For all the parts of this question, work must be shown to receive partial credit.

(a) Convert the hexadecimal integer A0 into its representation in base 5. (6 points)

(b) Convert the integer 212\_3 to octal. (6 points)

(c) Convert the decimal real number 49.3 into binary. (12 points)

(d) Compute the 2’s complement representation of the decimal integer \(-38\) using 8 bits. Give your answer in hexadecimal form. (9 points)

Question III (15 points total)

(a) Indicate the output produced by the following program. There are no syntax errors in the program. No explanation is needed. (6 points)

\[
\begin{verbatim}
#include <stdio.h>
int main(void) {
    int i = 97, j = 44, *p;
    p = &i; i++; printf("%d %d %d\n", i, j, *p);
    j = --(*p); printf("%d %d %d\n", i, ++j, *p);
    return 0;
}
\end{verbatim}
\]
(b) Indicate the output produced by the following program. There are no syntax errors in the program.

No explanation is needed. (9 points)

```c
#include <stdio.h>
void mystery(int *, int *, int *);

int main(void) {
    int p = -47, *q, r = 31;
    q = &r;
    printf("%d %d %d\n", p++, *q, --r);
    mystery(q, &r, &p);
    printf("%d %d %d\n", p, *q, r); return 0;
} /* End of main. */

void mystery (int *x, int *y, int *z) {
    *x = *y; *y = (*z)++; *z = 47;
    return;
} /* End of mystery */
```

**Question IV** (20 points)

Assume that the following type declaration is available:

```c
typedef struct node {
    int key; struct node *next;
} NODE, *PNODE;
```

Notice that each node of type NODE contains an integer key value and a pointer to the next node of a list. You are required to write a function compute with the following header:

```c
int compute (PNODE head)
```

Here, the parameter head points to the first node of a list each of whose nodes is of type NODE. The specifications for the function compute are as follows.

(a) If the list is empty, then the function must return the value 0.

(b) If the list has only one node, then function must return the key value stored in that node.

(c) If the list has two or more nodes, then function must return the sum of the key values stored in the last two nodes of the list.

Your need to show only the C code for the above function. You may use magic numbers and there is no need to include comments in your code.
Question V (20 points)

A company stores the information about each employee using the following structure.

```c
#define NAME_MAX 25
typedef struct employee {
    char name[NAME_MAX]; int age; float salary;
} EmpRec;
```

You are required to write the function `employee_stat` with the following header:

```c
float employee_stat(EmpRec company[], int nemp, int *age_range)
```

Here, the parameter `company[]` represents an array of records, with one record (of type `EmpRec`) corresponding to each employee. The parameter `nemp` represents the number of employees. (Thus, the indices for the `company[]` array vary from 0 to `nemp-1`.) Assume that the value of `nemp` is at least 1. At the end of the function, the reference parameter `*age_range` should have the number of employees whose age is at least 21 and at most 30. Further, the return value of the function must be the average salary of all the employees.

You need to show the code only for the above function. There is no need to include comments and you may use magic numbers in your code.