Spring 2014, CS288 Test 2, 2:30-3:45 pm, Fri, 4/4/2014, GITC 1400

Name:

The exam has 5 pages. Do not take any page(s) with you. Any missing page(s) will result in failure in the exam. This exam is closed book close notes. Do not exchange anything during the exam. **No questions will be answered during the exam.** If you are in doubt, state your assumptions below, including typos if any.

ANSWERS

I have read and understood all of the instructions above. On my honor, I pledge that I have not violated the provisions of the NJIT Academic Honor Code.

Signature: _____ **Date:** _____

Answers to Questions 1 to 13 (3 points each) = 39 points + 1

1	2	3	4	5	6	7	8	9	10	11	12	13

Questions 1-3: Assume radix sorting of 1024 floats on a 32-bit machine with 8 passes (rounds). The floats are initially stored in lst[1024] and the sorted floats will be available in lst at the end of sorting. float buf[1024] is available as working space.

1. What is the number of buckets?
   1<<(b/p) ????

2. The bit mask in *hexadecimal* is?
   (1<<(b/p))-1 ????

3. Find the number of data assignments for correcting the result when completed. For example, moving lst[i] to buf[j], or buf[j]=lst[i]; is a data assignment.

4. Given `float f;` which of the C statements would allow you to access the binary equivalent of `f`:
   a) &f
   b) *f
   c) (unsigned long *) (&f)
   d) (unsigned long *) (*f)
   e) * (unsigned long *) (&f)

5. From problem 4, you now have the floating point number f converted to x. Assuming char s[32]; int i,n=32; which of the following C statements would store the binary equivalent of x in the string s, where s[0] holds the sign bit (the most significant bit) of the original number f while s[31] holds the least significant bit of the original number f:

```
a) for (i=0;i<n;i++) { s[n-1-i] = "01"[x|1]; x = x + 1; }
b) for (i=0;i<n;i++) { s[n-1-i] = "10"[x&1]; x = x + 1; }
c) for (i=0;i<n;i++) { s[n-1-i] = "01"[x&1]; x = x >> 1; }
d) for (i=0;i<n;i++) { s[n-1-i] = "10"[x&1]; x = x >> 1; }
e) none of the above
```

6. Given `char *str = "a?,?b,,??c#,,";` what would `strtok(str, "?");` return?

7. Continuing Problem 6 write fill in the blank such that strtok() will return `b`.

```
strtok(_____________________________________________);
```

8. Continuing Problem 7, what would `strtok(NULL, "?#,");` return?

For 9-13 on the 15-puzzle state-space search consider succ has three nodes (p,q,r) while open has two nodes (x,y).

9. What search strategy would result in open=(p,q,r,x,y) after merging succ and open?
   a)depth b)breadth c)best d)branch-bound e)a*

10. What search strategy would result in open=(x,y,p,q,r) after merging succ and open?
    a)depth b)breadth c)best d)branch-bound e)a*

11. Depth first search relies solely on
    a)g b)h c)f=func(g,h) d)any 2 of f,g,h e)can't determine

12. Intelligent heuristic search such as A* relies on
    a)g b)h c)f=func(g,h) d)any 2 of f,g,h e)can't determine

13. What is the branching factor for the 15-Puzzle problem?
    a)2 b)3 c)4 d)5 e)can't determine

Problem 14 (splitting string - 20 points): Write a C function that splits a string *line* separated by commas and stores the values in an array of strings *fields*. The number of commas is *unknown* and your function must be able to handle any number of commas in the string. Use the following built-in functions: strtok(line, delim); strtok(NULL, delim); malloc(strlen(token)); strcpy(fields[i],token);

/* at the end of this function, fields will have n strings stored */

```
void split_line(char **fields,char *line) {
  int i=0;
  char *token, *delim;
```

```
delim = ","
field [i] = strtok(line, delim)
while (fields [i] ) {
        fields[++i] = strtok(num, delim)
}
or
delim = ","
token = strtok(line, delim)
while (token != NULL) {
        field[i] = (char*)malloc(strlen(token))
        strcpy(list[i], token)
        token = strtok(NULL, delim
        i++
}
```

```
}
```

Problem 15 (Linked list - 20 points): Write a C function () which creates a struct clip, sets an integer passed as parameter to views, and inserts the newly created struct in descending order of views to the list. *head* points to the first clip in the list. Your function must be able to handle any number of clips, including none in the beginning.

```
#include <stdio.h>
void insert();
struct clip { int views; struct clip *next; } *head;
void main() {
   struct clip *head; int n;
   append(&head,1);
   n = length(head);
}

void insert(_____________________________________) {
   struct clip *cp,*tp;









}
```

Problem 16 (*Signed* integer radix sort - 20 points): Write a C program for sorting 32-bit signed integers using radix sort with a group of 8 bits. Use the variables listed below. Assume lst is initialized with n numbers.

```
#define N 1048576
#define BIN 256
#define MAXBIT 32
#define LST 1
#define BUF 0

int n,group,bin;
int flag; /* to show which one holds numbers: lst or buf */
int lst[N],buf[N];
int count[BIN], map[BIN], tmap[BIN];

int main(int argc, char **argv){
    int i;
    flag = LST;
    initialize(); /* initialize lst with n random floats */
    for (i=0;i<MAXBIT;i=i+group) radix_sort(i); /* move lst to buf or buf to lst depending on the iteration number */
    correct(); /* sorted numbers must be in lst */
}
void radix_sort(int idx) {
    int i,j,k,mask; /* initialize mask for lifting the 4 least significant bits. */
    int *src_p,*dst_p; /* cast lst and buf to int pointers to treat lst/buf as int's */
    /* set src_p and dst_p*/




    /* count */

```

SEE FALL 2012 FOR SOLUTION

```
    /* map */




    /* move */




}

void correct() {




}
```