

Fall 2014, CS288 Test 2, 11:30-12:45 pm, Thur, 11/6/2014, GITC 1400

Name: Sarp Uslu

Make sure you have five 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. You all have the same exam. **No questions will be answered during the exam, including typos.** I don't want to give different answers to different people. If you are in doubt, briefly state your assumptions below, including typos if any. Make sure you are sitting where you are assigned. **You may not go to the restroom.**

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:

Problems 1-2: For radix sorting of y signed integers on a z -bit machine, radix sort requires x passes (rounds), where x , y , and z are power of 2 numbers.

1. (5 points) Find the bit mask in terms of x , y , and z .

$$\left(1 \ll \frac{z}{x}\right) - 1$$

2. (5 points) Find the number of buckets in terms of x , y , and z .

$$1 \ll \frac{z}{x}$$

3. (5 points) Given float y ; write a C statement to access the binary equivalent of y .

-2
~~(unsigned int *)~~ (By Long -2)

4. (5 points) From the above problem, you now have the floating point number y converted to z . Given char $x[32]$; int $i, n=32$; write three C statements to store the binary equivalent of z in the string x , where $x[0]$ holds the sign bit (the most significant bit) of the original number y while $x[31]$ holds the least significant bit of the original number y :

for ($i=0; i < n; i++$) {
 $x[n-1-i] = (z \gg i)$
 $z \gg 1;$

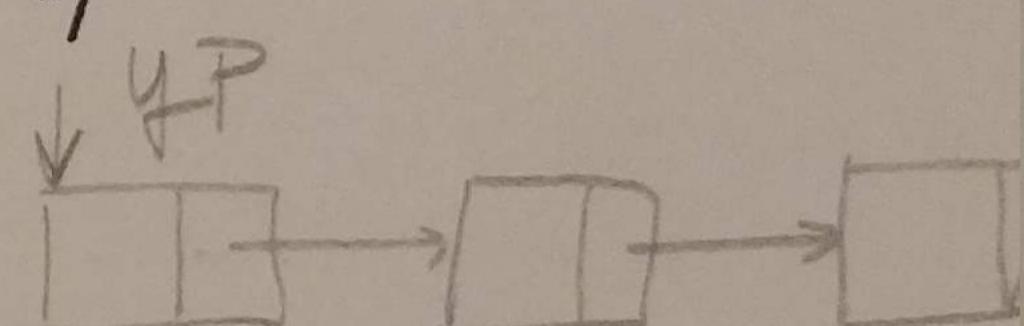
3

5. (10 points) Write four C statements to complete split_line() function that splits a string separated by commas and stores in fields. Your function must be able to handle any number of commas. Use strtok(). Define delim.

```
void split_line(char **fields, char *line) {  
    int i=0; char *delim;  
    delim = ",";  
    fields[i] = strtok(line, delim);  
    while (fields[i] != null) {  
        fields[++i] = strtok(null, delim);  
    }  
}
```

6. (10 points) Write seven C statements to complete a C function that inserts a clip xp at the end of the list pointed by yp.

```
void append(struct clip *yp, struct clip *xp) { /* yp=head pointer, xp=new pointer, */  
    struct clip *zp; /* zp=tmp clip pointer */  
  
    xp = malloc(sizeof(clip));  
    xp->next = null;  
    zp = yp;  
    while (zp->next != null) {  
        zp = zp->next;  
    }  
    zp->next = xp;
```



Problem 7 (Floating Point Radix sort - 30 points): Fill in radix_sort() below to complete a C program for sorting 32-bit floating point numbers using x-bit radix sort. Use the variables listed below. DO NOT use any other variables.

```
#define N 1048576
#define BIN 256
#define MAX 32

int n,x,mask,bucket;
float lst[N],buf[N];
int cnt[BIN], map[BIN], tmap[BIN];

int main(int argc, char **argv){
    int i;
    initialize(argc,argv);
    for (i=0;i<MAX;i=i+group) /* initialize all the global variables listed above */
        radix_sort(i); /* group is the number of bits or radix used in each round. */
    correct(); /* not considered in this exam */
}

void radix_sort(int idx) { /* at the end of this function, numbers in lst will have moved to buf */
    int i,j;
    /* assume cnt is initialized to zeros */

    /* (10 points) write TWO C statements for counting. Use i for loop iteration numbers */
    /* for(i=0; i<N; i++) {
        cnt[lst[i] & mask]++;
    } */

    /* (10 points) write THREE C statements for mapping. Use i for loop iteration numbers */
    j=0;
    map[0]=0;
    for(i=1; i<BIN; i++) {
        map[i]=map[i-1]+cnt[i-1];
    }

    /* (10 points) write TWO C statements for moving lst to buf. Use i for loop iteration numbers */
    for(i=0; i<N; i++) {
        buf[map[lst[i] & mask]] = lst[i];
    }
}
```

board[i][j]

Problems 8-10 (Fifteen Puzzle - 30 points)

```
struct node {
    int board[4][4]; struct node *next;
};
```

struct node *x, *y, *initial, *goal, *succ, *open, *closed;

Assume initial- and goal states are initialized.

22

Problem 8 (10 points): Write a C function nodes_same(x,y) that returns 1 if two nodes x and y are same. Otherwise, 0.

```
int nodes_same(struct node *x, struct node *y) {  
    int i, j;  
    for(i=0; i<4; i++) {  
        for(j=0; j<4; j++) {  
            if((x->board[i][j]) != (y->board[i][j])) {  
                return 0; }  
        }  
    }  
    return 1; }
```

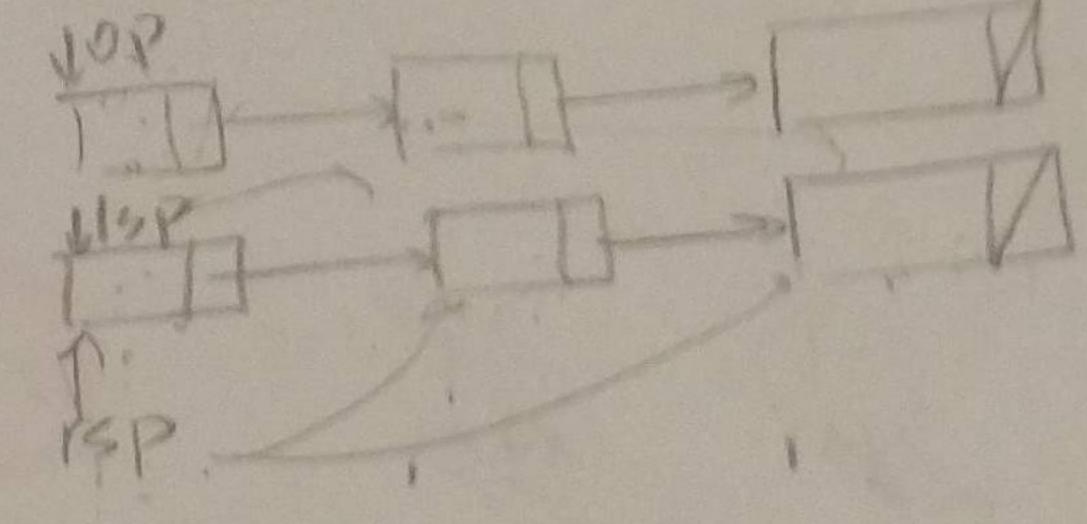
10

Problem 9 (10 points): Write a C function goal_found(succ,goal) that returns 1 if goal is in succ. Otherwise, 0. succ is a pointer to a singly linked list of nodes. See the definition of variables shown above. Use nodes_same() you wrote above.

```
int goal_found(struct node *succ, struct node *goal) {  
    if(succ == null) { return 0; }  
    while(1) {  
        if(nodes_same(succ, goal) == 1) {  
            return 1; }  
        else {  
            if(succ->next != null) {  
                succ = succ->next; }  
            else {  
                return 0; } // end of list. goal not found  
        }  
    }  
}
```

9

why do this? *(with arrows pointing from the question to the code)*



Problem 10 (10 points): Write a C function filter(open,succ) that returns a pointer to succ, a singly linked list of nodes that are not in open. Use nodes_same(x,y) you wrote in the previous question. **Use only the variables listed below.**

```
struct node *filter(struct node *open, struct node *succ){  
    struct node *lsp, *rsp;  
    struct node *op;
```

```
    lsp = succ;  
    rsp = succ;  
    op = open;  
    while (op->next != null) {  
        while (rsp->next != null) {  
            if (nodes_same(op, rsp)) {  
                lsp = rsp->next;  
            }  
            rsp = rsp->next;  
        }  
        op = op->next;  
    }  
    return succ;
```

3