Computer Science Foundation Exam

August 2, 2002

COMPUTER SCIENCE I

Section I A

No Calculators!

Name: KEY

SSN: ______________________

Score: ¤ **50**

In this section of the exam, there are four (4) problems

<u>You must do all of them.</u>

The weight of each problem in this section is indicated with the problem. The algorithms in this exam are written in C programming language notation. Partial credit cannot be given unless all work is shown.

As always, be complete, yet concise, and above all <u>be neat</u>. Credit cannot be given when your results are unreadable.

1. **(10 points)**
   What is the exact output from the following program?

```
#include <stdio.h>

int  q1 (int *, int, int *);

int main ( ) {
     int a = 3;
     int b = 15;
     int *c;

     c = &b;
     a = q1(&a, b, c);
     printf(“2. a = %d, b = %d, *c = %d\n”, a, b, *c);
     return 0;
}

int q1 (int *x, int y, int *z){
     int a = 7;
     int *p;

     printf(“1. *x = %d, y = %d, *z = %d\n”, *x, y, *z);
     p = z;
     *p = a + *x;
     return (*x + *z + y);
}
```

Output is:

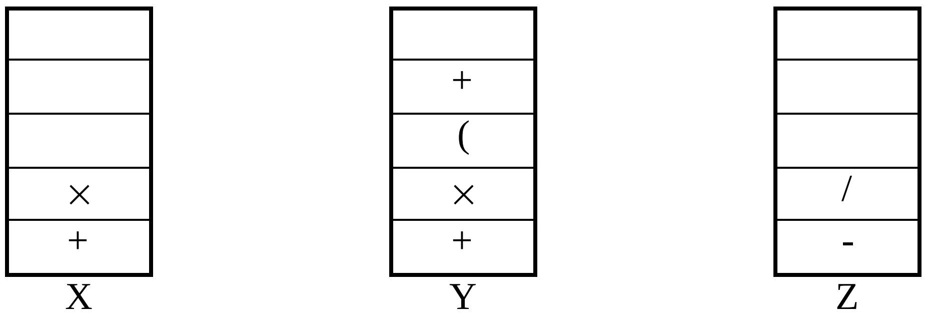
```
1.  *x = 3, y = 15, *z = 15

2.  a = 28, b = 10, *c = 10
```

2. (15 points – 9pts(a), 6pts(b))

(a) Consider the following expression in infix notation:

$$A + B \times^{X} (2 + C^{Y}) - D /^{Z} E$$

Using a stack, transform this infix expression into a postfix expression. Trace the state of the operator stack as each character of the infix expression is processed. Show the contents of the operator stack at the indicated points in the infix expression (points X, Y and Z). Put the final postfix expression in the box.

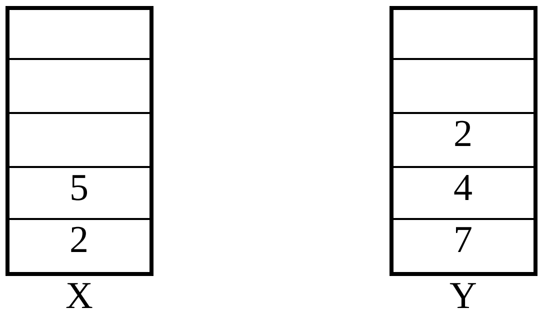


The resulting postfix expression is: A B 2 C + * + D E / –

(b) Given the postfix expression shown below, use a single stack to evaluate the expression. Show the contents of the stack at the reference points marked in the expression as X and Y. Assume the variables have the following values at the time of the evaluation:

A = 2, B = 3, C = 6, D = 2, E = 2

The postfix expression: $A\ 2\ B\ +^{X} \times C\ D\ / - 4\ E^{Y} \times +$



Final value of the expression is: 15

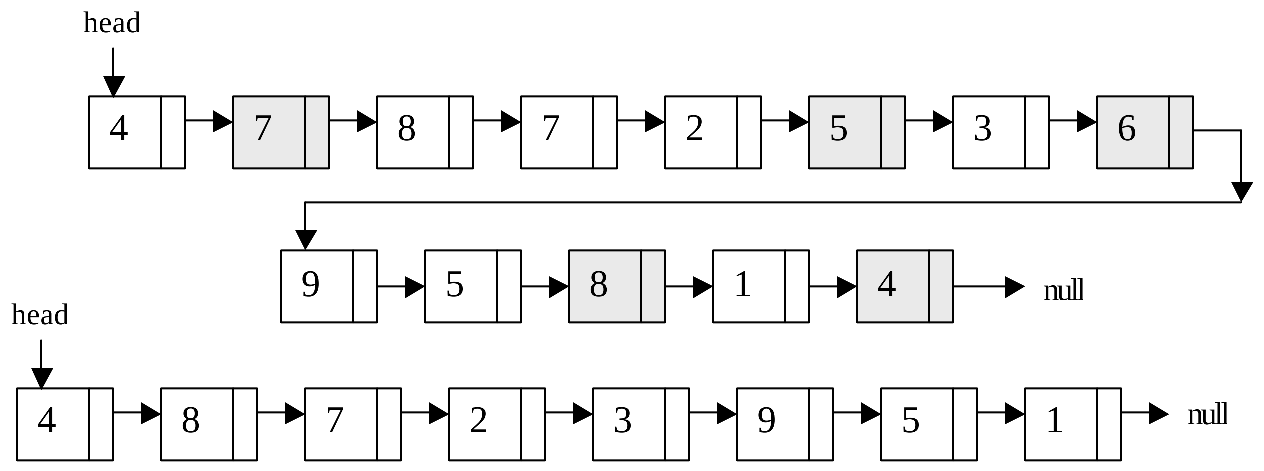
3. (10 points)

Trace the effect of executing the code segment shown below on the linked list structure, also shown below, by drawing the linked list as it would look after the code has been executed.

The linked list:

```
struct node{
   int data;
   struct node *next;
};

p = head;
while (p != NULL)
{  if (p->data <= 5)
   { newp = (struct node *) malloc(sizeof(struct node));
     new->data = p->data + 3;
     newp->next = p->next;
     p->next = newp;
     p = newp;
   }
   else
     p = p->next;
}
```

Solution: The code adds a new node after every node whose data value is less than or equal to 5. The new node has a data value equal to its predecessor plus 3. (New nodes are shaded.)



4. (15 points – 5 points each)

Answer each of the following "timing" questions concerning an algorithm of a particular order and a data set of a particular size. Assume that the run time is affected only by the size of the data set and not its composition and that **n** is an arbitrary integer.

(a) An algorithm known to be O(n $\log_2$ n) requires 64 seconds to solve a problem instance of size n = 32. If the same algorithm is used to solve a different problem instance and the time required was 256 seconds, what size was this other problem instance?

$$\frac{n_{old}}{t_{old}} = \frac{n_{new}}{t_{new}} \Rightarrow \frac{(32\log_2 32)}{64} = \frac{n}{256} \Rightarrow n_{new} = \frac{32(5)(256)}{64} = 640$$

(b) An algorithm known to be O(n^3) requires 50 seconds to solve a problem instance of size n = 16. To the nearest second, how long will it take the algorithm to solve a problem instance of size n = 20?

$$\frac{n_{old}}{t_{old}} = \frac{n_{new}}{t_{new}} \Rightarrow \frac{16^3}{50} = \frac{20^3}{t_{new}} \Rightarrow t_{new} = \frac{50(20)^3}{16^3} = 97.6 = 98\,\text{sec}$$

(c) Answer questions (i) and (ii) for the C code segment shown below.

```
x = 0;
for (i=1; i<=2*n; i++)
    for(k=1; k<=3*(n+1); k++)
         x = x + k;
```

(i) What is the Big-Oh order of the code segment?

$$\sum_{i=1}^{2n}\sum_{k=1}^{3n+3} 1 = 2n(3n+3) = 6n^2 + 6n = O(n^2)$$

(ii) What will the value of **x** be when the loops end?

$$\sum_{i=1}^{2n}\sum_{k=1}^{3n+3} k = 2n\frac{(3n+3)(3n+4)}{2} = 2n\frac{9n^2+21n+12}{2} = \frac{18n^3+42n^2+24n}{2}$$

$$= 9n^3 + 21n^2 + 12n$$