COP 4530 Term Exam 4
Fall 2001
Professor Chris Lacher

Name: ____________________
SSN: ____________________
CS Username: ____________________
Score: ____________________ /100

This test contains 2 questions (10 subquestions) on 3 pages. Each subquestion is worth 10 points.

1. Suppose that a hash table is (1) implemented using a private bucket vector object declared as `TVector < TList < TAssociation < String, int > > > bv;` (2) table insert uses `TList<>::PushBack();` and (3) `hash_function(String S)` is implemented by:

```
unsigned int hash_function (const String& S)
{
  unsigned int hval(0), i;
  for (i = 0; i < S.Size(); ++i)
    hval += S[i] - 'a';
  return hval;
}
```

hash function vals	
String	value
a	0
b	1
c	2
d	3
e	
f	
g	
h	
i	
j	
k	
l	
m	
n	
o	
p	
q	
r	
s	
t	
u	
v	
w	
x	
y	
z	

Begin with a 5-bucket table `t` after the operations

```
t.Insert("ab",1);
t.Insert("cd",2);
t.Insert("ef",3);
t.Insert("gh",4);
t.Insert("pq",5);
```

have been performed.

a. Illustrate the result of the operation:
`t.Dump(cout);`

b. Illustrate the result of the traversal (`I` is a table iterator):
`for (I = t.Begin(); I != t.End(); ++I) cout << *I;`

Now assume that these five *additional* operations

```
t.Remove("ab");
t.Remove("gh");
t.Insert("ab",6);
t.Insert("jk",7);
t.Insert("pq",8);
```

have been performed.

c. Illustrate the result of the operation:

```
t.Dump(cout);
```

d. Illustrate the result of the traversal (`I` is a table iterator):

```
for (I = t.Begin(); I != t.End(); ++I) cout << *I;
```

e. Under certain assumptions, a hash table has expected runtime O(1) for its insert, remove, and look-up operations. What are these assumptions?

2. An implementation of priority queue uses a vector `v` of elements of type `T` and a comparison predicate object `LessThan` for type `T`. The priority queue `Push()` and `Pop()` methods use the generic algorithms g_`push_heap()` and `g_pop_heap()` which are derived using a heap structural model superimposed onto `v`. A *heap* is a complete binary tree with a certain order property.

   a. *Name* the order property used to define a heap.

   b. *Define* the order property used to define a heap.

   In the following, the type `T` is `char` and `LessThan` is such that characters with lower ascii value have higher priority (e.g., `'a'` has higher priority than `'b'`).

   c. Begin with `v = [d,f,k,m,t,p]`. Show the binary tree structure we impose on `v`. Is this a heap? (Explain your answer.)

   d. Beginning with the *tree in part c above*, show each stage of the binary tree structure before, during, and after `Push(b)`. Also, illustrate the final state of `v`.

      `v = [`

   e. Beginning with the *original tree in part c*, show each stage of the tree before, during, and after `Pop()`. Also, illustrate the final state of `v`.

      `v = [`