Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

COM 205 Data Structures
Midterm Exam, 13.11.2014, 10:00, D-1, D-8
Time : **100** Minutes

NUMBER : ..... **NAME :** .....

Rules to be Obeyed During the Exam **SIGNATURE :** .....

EXAM GRADE	
[.....]	.....

1. Cell phones are not allowed to be used as a calculator or a watch. They must be switched off and placed in the pocket.
2. Brief information about the exam will be given at the begining, then no one is not allowed to ask a question during the exam.
3. Do not to forget to sign this paper after writing your number and name.

```
DoublyLinkedList* fList(DoublyLinkedList* list1)
{
  DoublyLinkedList* list2 = new DoublyLinkedList();
  DoublyNode* nodeA = NULL;
  DoublyNode* nodeB = NULL;

  while (!list1->empty())
  {
    nodeA = list1->header->next;
    nodeB = list1->header->next->next;

    while (nodeB != list1->trailer)
    {
      if (nodeB->score > nodeA->score)
      {
        nodeA = nodeB;
        nodeB = nodeB->next;
      }
      else
        nodeB = nodeB->next;
    }

    list2->addBack(nodeA->elem, nodeA->score);
    list1->remove(nodeA);
  }

  return list2;
}

void main()
{
  DoublyLinkedList* list1 = new DoublyLinkedList();
  list1->addFront("Paul", 720);
  list1->addFront("Rose", 590);
  list1->addFront("Jack", 510);
  list1->addFront("Anna", 660);
  list1->addFront("Rob", 750);

  DoublyLinkedList* list2 = fList(list1);
  list2->printH2T();
}
```

1. What does **fList()** do? Explain your answer.**(30P)**

```
int biC(int n, int k)
{
   if (k == 0) return 1;
   if (k == n) return 1;
   return biC(n - 1, k-1) + biC(n - 1, k);
}

void main()
{
   for (int n = 0; n < 5; n++)
   {
      for (int k = 0; k <= n; k++)
      {
         cout << biC(n, k) <<" ";
      }

      cout << endl;
   }
}
```

2. What is the output of the program above? **(30P)**

```
bool empty()
{
   return (header->next == trailer);
}

void addFront(const string& e, const int& i)
{
   add(header->next, e, i);
}

void add(DoublyNode* v, string& e, int& i)
{
   DoublyNode* u      = new DoublyNode;
   u->elem            = e;
   u->score           = i;
   .....
   .....
   .....
   .....
}

void printH2T()
{
   if (empty())
   {
      cout << "List is empty !" << endl;
      return;
   }

   DoublyNode* first = header;
   while (!(first->next == trailer))
   {
      cout << first->next->elem  <<
      "\t" << first->next->score << endl;
      first = first->next;
   }
}

void main()
{
   DoublyLinkedList list;
   list.addFront("Rob",  750);
   list.addFront("Paul", 720);
   list.printH2T();
}
```

3. Taking into account the lines represented by `.....` in the function **add()** answer the following choices :

i) (10P) *(You'll loose 5Ps from each wrong answer)*

If the lines are like these

```
u->prev          = v->prev;
u->next          = v;
v->prev->next    = u;
v->prev          = u;
```

the `printH2T()` function :

(A) will print the list elements.
(B) will print `"List is empty !"`.
(C) will enter into an infinite loop.

ii) (10P) *(You'll loose 5Ps from each wrong answer)*

If the lines are like these

```
u->next          = v;
v->prev          = u;
v->prev->next    = u;
u->prev          = v->prev;
```

the `printH2T()` function :

(A) will print the list elements.
(B) will print `"List is empty !"`.
(C) will enter into an infinite loop.

iii) (10P) *(You'll loose 5Ps from each wrong answer)*

If the lines are like these

```
u->prev          = v->prev;
u->next          = v;
v->prev          = u;
v->prev->next    = u;
```

the `printH2T()` function :

(A) will print the list elements.
(B) will print `"List is empty !"`.
(C) will enter into an infinite loop.

iv) (10P) *(You'll loose 5Ps from each wrong answer)*

If the lines are like these

```
v->prev->next    = u;
u->next          = v;
v->prev          = u;
u->prev          = v->prev;
```

the `printH2T()` function :

(A) will print the list elements.
(B) will print `"List is empty !"`.
(C) will enter into an infinite loop.