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

```cpp
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();
}
```

2. What is the output of the program above? (30P)
3. Taking into account the lines represented by ..... in the function add() answer the following choices:

i) (10P)  (You’ll lose 5Ps from each wrong answer)
If the lines are like these

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

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 lose 5Ps from each wrong answer)
If the lines are like these

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

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

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

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 lose 5Ps from each wrong answer)
If the lines are like these

```cpp
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;
    }
}
```

```cpp
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;
    }
}
```

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 lose 5Ps from each wrong answer)
If the lines are like these

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

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

the printH2T() function:
(A) will print the list elements.
(B) will print "List is empty !".
(C) will enter into an infinite loop.