


```

void insertOrdered(DoublyNode* newNode,
                  DoublyNode* current)
{
    if(..... || .....)
    {
        newNode->next      = current->next;
        newNode->prev      = current;
        current->next->prev = newNode;
        current->next      = newNode;
    }
    else
        insertOrdered(newNode, current->next);
}

int main()
{
    DoublyLinkedList list; DoublyNode* newNode;

    newNode = new DoublyNode;
    newNode->elem = "Paul";  newNode->score = 720;
    list.insertOrdered(newNode, list.header);

    newNode = new DoublyNode;
    newNode->elem = "Rose";  newNode->score = 590;
    list.insertOrdered(newNode, list.header);

    newNode = new DoublyNode;
    newNode->elem = "Anna";  newNode->score = 660;
    list.insertOrdered(newNode, list.header);

    newNode = new DoublyNode;
    newNode->elem = "Mike";  newNode->score = 1105;
    list.insertOrdered(newNode, list.header);
}

```

3. insertOrdered() fonksiyonunu tamamlayınız. (25P)

Not → header ve trailer'ın score değerini 0 varsayınız.

Yanlış cevaptan 5P kılacaktır.

- (A) `if ((current == trailer) || (newNode->score <= current->score))`
- (B) `if ((current->next == trailer) || (newNode->score <= current->score))`
- (C) `if ((current == trailer) || (newNode->score <= current->next->score))`
- (D) `if ((current->next == trailer) || (newNode->score <= current->next->score))`

```

void strings(string str, int i, int n)
{
    if (i == n - 1)
    {
        cout << str << endl;
        return;
    }

    for (int j = i; j < n; j++)
    {
        swap(str[i], str[j]);
        strings(str, i + 1, n);
        swap(str[i], str[j]);
    }
}

void main()
{
    string str = "NEO";
    strings(str, 0, str.length());
}

```

4. Yukarıdaki programın çıktısı nedir?

(25P)