

## Datorsystemteknik LABORATION 2

1)

Skapa enkellänkad lista enligt exempel på fl9 (clist.c, enkel\_list.c)

Skriv funktioner för:

- Skapa ny nod i listan (allokera minne för elementen)
- Skriv ut lista
- "Insert" av ny nod i listan så att listan alltid är sorterad ( efter nInfo )
- "Delete" av första noden i listan

Testa funktionerna genom att sätt in 10 element i listan som är initierade med:

```
rand() % 100
```

Detta ger ett slumpstal i området 0 till 99, % betyder modulus (dvs resten vid heltalsdivision), inkludera math.h för modulus samt stdlib.h för `rand()`. Rand-funktionen kan verka icke slumpmässig, se `srand()` för initering av frö till slumpstalsgeneratören.

2)

Gör om ovanstående list till en dubbellänkad lista med hjälpnoderna head och tail.

Noderna definieras som:

```
struct DListNode {
    int nInfo;
    struct DListNode *prev;
    struct DListNode *next
};
```

Listan definieras som:

```
struct DList {
    struct DListNode *head;
    struct DListNode *tail;
};
```

3)

Utvidga ovanstående så att noderna i listan innehåller en dynamiska datastruktur.

```
struct DListNode {
    MsgObject *prMsgObj;
    struct DListNode *prev;
    struct DListNode *next
};
```

där MsgObject är definerad som:

```
typedef _MsgObject {
    int nID;
    int nMsgType; /* 0 for standard message, 1 for priority
message */
} MsgObject;
```

Initiera noder pss som ovan med `rand() % 100` samt för `nMsgType` `rand() % 2`

Skriv funktion för:

- print all
- print standard message
- print priority message

Att tänka på!

- Vid allokering av noder i listan måste även plats för ett `MsgObject` allokeras.