Beispiel 1 (60 Punkte): ADT „Liste“

Implementieren Sie den abstrakten Datentyp „Einfach verkettete Liste“ (singly linked list) als Klasse. Verwenden Sie dafür das unten gegebene Interface.

Beispiel 2 (40 Punkte): Dynamic Array

Implementieren das Konzept von „dynamic arrays“. Entwickeln Sie ein dazu Modul **`dynamic_array`** und ein Testmodul, und verwenden Sie auch dafür das unten gegebene Interface.

Anmerkungen: (1) Geben Sie Lösungsideen an. (2) Strukturieren und arbeiten Sie sauber. (3) Kommentieren Sie ausführlich. (4) Geben Sie ausreichend Testfälle ab. (5) Prüfen Sie alle Eingabedaten auf ihre Gültigkeit.

```
class slist {
  public:
    typedef /*...*/ value_t;
    typedef void (* function_t) (value_t &);

    slist ();
    slist (slist const & src);
   ~slist ();

    void apply (function_t f);
    bool at (std::size_t n, value_t & value);
    std::size_t clear ();
    bool contains (value_t const & value) const;
    std::size_t count (value_t const & value) const;
    bool empty () const;
    bool equal (slist const & rhs) const;
    void insert_sorted (value_t const & value);
    bool pop_back (value_t & value);
    bool pop_front (value_t & value);
    std::ostream & print (std::ostream & out =
          std::cout) const;
    void push_back (value_t const & value);
    void push_front (value_t const & value);
    std::size_t remove_all (value_t const & value);
    std::size_t size () const;

  private:
    struct node_t {
     // ...
    };
  // ...
};
```

```
namespace dyn{

  typedef std::string    data_type;
  //typedef data_type * element_type;

  class dynamic_array_type {

  private:
    //element_type *    m_table;
    data_type **        m_table;
    size_t              m_rows;
    static const size_t m_cols = 20;
    size_t m_max_size;

  public:

    dynamic_array_type();
    dynamic_array_type(size_t const size);
    ~dynamic_array_type();

    void clear();
    void initialize(size_t const size);

    size_t get_max_size() const;

    data_type & at(size_t const pos);
    data_type   get(size_t const pos) const;
    bool        set(data_type const & data,
                    size_t const pos);

    void print(std::ostream & out) const;
  };
}
```