建立单链表的常用方法

解答：

建立单链表的常用方法有两种。下面以顺序存储为例来叙述。

(1)　头插法建表

该方法从一个空表开始，读取数组 a 中的字符，生成新结点，将读取的数据存放到新结点的数据域中，然后将新结点插入到当前链表的表头上，直到结束为止。算法如下：

```
void CreateListF(Snode *&L, ElemType a[], int n)
{     Snode *s; int i;
      L = (Snode *) malloc(sizeof(Snode));
      L->next = NULL;
      for (i=0; i<n;i++)
      {   s = (Snode *)malloc(sizeof(Snode));
          s->data = a[i];
          s->next = L->next;
          L->next = s;
      }
}
```

(2)　尾插法建表

头插法建立链表虽然算法简单，但生成的链表中结点的次序和原数组元素的顺序相反，若希望两者次序一致，可采用尾插法。该方法是将新结点插到当前链表的表尾上，为此必须增加一个尾指针 r,使其始终指向当前链表的尾结点。算法如下：

```
void CreateListR(Snode *&L, ElemType a[], int n)
{     Snode *s, *r; int i;
      L = (Snode *) malloc(sizeof(Snode));
      L->next = NULL;
      r = L;
      for (i=0; i<n;i++)
      {   s = (Snode *)malloc(sizeof(Snode));
          s->data = a[i];
          r->next = s;
          r = s;
      }
      r-> next = NULL;
}
```

问题：

建立单链表的常用方法

解答：

建立单链表的常用方法有两种。下面以顺序存储为例来叙述。

(1)　头插法建表

该方法从一个空表开始，读取数组a中的字符，生成新结点，将读取的数据存放到新结点的数据域中，然后将新结点插入到当前链表的表头上，直到结束为止。算法如下：

```
void CreateListF(Snode *&L, ElemType a[], int n)
{     Snode *s; int i;
      L = (Snode *) malloc(sizeof(Snode));
      L->next = NULL;
      for (i=0; i<n;i++)
      {   s = (Snode *)malloc(sizeof(Snode));
          s->data = a[i];
          s->next = L->next;
          L->next = s;
      }
}
```

(2)　尾插法建表

头插法建立链表虽然算法简单，但生成的链表中结点的次序和原数组元素的顺序相反，若希望两者次序一致，可采用尾插法。该方法是将新结点插到当前链表的表尾上，为此必须增加一个尾指针r,使其始终指向当前链表的尾结点。算法如下：

```
void CreateListR(Snode *&L, ElemType a[], int n)
{     Snode *s, *r; int i;
      L = (Snode *) malloc(sizeof(Snode));
      L->next = NULL;
      r = L;
      for (i=0; i<n;i++)
      {   s = (Snode *)malloc(sizeof(Snode));
          s->data = a[i];
          r->next = s;
          r = s;
      }
      r-> next = NULL;
}
```