

实验11

1 (10分)

编写函数deln，具有删除链表中第n个结点的功能。再编写主函数，按输入顺序建立不带头结点的职工信息单链表，然后调用del函数删除某个职工的信息，并输出删除后的职工信息表中的职工信息。假设链表结构如下：

```
struct staff

{

    char  num[6];    //职工工号

    char  name[20];  //职工姓名

    double wage;     //职工工资

};
```

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  // 定义职工信息结构体
6  struct staff {
7      char num[6];        // 职工工号
8      char name[20];      // 职工姓名
9      double wage;        // 职工工资
10     staff *next;        // 指向下一个结点的指针
11 };
12
13 // 删除链表中第n个结点
14 void deln(staff *&head, int n) {
15     if (head == nullptr || n <= 0) {
16         cout << "Invalid position." << endl;
17         return;
18     }
19
20     staff *temp = head;
21
22     // 如果删除的是第一个结点
23     if (n == 1) {
24         head = temp->next;
```

```

25     delete temp;
26     return;
27 }
28
29 // 找到第n-1个结点
30 for (int i = 1; temp != nullptr && i < n - 1; i++) {
31     temp = temp->next;
32 }
33
34 // 如果第n个结点不存在
35 if (temp == nullptr || temp->next == nullptr) {
36     cout << "Position exceeds the list length." << endl;
37     return;
38 }
39
40 // 删除第n个结点
41 staff *next = temp->next->next;
42 delete temp->next;
43 temp->next = next;
44 }
45
46 // 输出职工信息
47 void printList(staff *head) {
48     staff *temp = head;
49     while (temp != nullptr) {
50         cout << "职工工号: " << temp->num << ", ";
51         cout << "职工姓名: " << temp->name << ", ";
52         cout << "职工工资: " << temp->wage << endl;
53         temp = temp->next;
54     }
55 }
56
57 // 主函数
58 int main() {
59     staff *head = nullptr, *tail = nullptr;
60
61     int n;
62     cout << "请输入职工人数: ";
63     cin >> n;
64     cin.ignore(); // 忽略多余的换行符
65
66     // 按顺序输入职工信息
67     for (int i = 0; i < n; i++) {
68         staff *newStaff = new staff;
69         cout << "请输入第 " << i + 1 << " 个职工的工号: ";
70         cin >> newStaff->num;
71         cin.ignore();

```

```

72     cout << "请输入第 " << i + 1 << " 个职工的姓名： ";
73     cin.getline(newStaff->name, 20);
74     cout << "请输入第 " << i + 1 << " 个职工的工资： ";
75     cin >> newStaff->wage;
76     newStaff->next = nullptr;
77
78     if (head == nullptr) {
79         head = newStaff;
80         tail = newStaff;
81     } else {
82         tail->next = newStaff;
83         tail = newStaff;
84     }
85 }
86
87 // 输出原链表中的职工信息
88 cout << "\n原职工信息表:" << endl;
89 printList(head);
90
91 // 输入删除的职工位置
92 int pos;
93 cout << "\n请输入要删除的职工位置： ";
94 cin >> pos;
95
96 // 删除链表中第pos个职工
97 deln(head, pos);
98
99 // 输出删除后的职工信息
100 cout << "\n删除后的职工信息表:" << endl;
101 printList(head);
102
103     system("pause");
104     return 0;
105 }

```

说明：

1. **职工结构体 `staff`**：包含职工工号、姓名、工资和指向下一个职工的指针 `next`。
2. **`deln` 函数**：该函数删除链表中第n个职工的信息。
 - 如果删除的是链表的第一个结点，则直接更新头结点。
 - 如果删除的是链表中的中间或尾部结点，则先找到第n-1个结点，再修改其 `next` 指针，删除第n个结点。
3. **`printList` 函数**：用于遍历链表并输出每个职工的信息。

4. 主函数 `main`：首先创建职工链表，按顺序输入职工信息。然后通过调用 `deln` 函数删除指定位置的职工，并输出删除后的职工信息。

D:\2024\课内\C++\第十一章\11.1\Debug\11.1.exe

```
请输入职工人数：2
请输入第 1 个职工的工号：1
请输入第 1 个职工的姓名：2
请输入第 1 个职工的工资：1111
请输入第 2 个职工的工号：2
请输入第 2 个职工的姓名：23
请输入第 2 个职工的工资：1111

原职工信息表：
职工工号：1, 职工姓名：2, 职工工资：1111
职工工号：2, 职工姓名：23, 职工工资：1111

请输入要删除的职工位置：2

删除后的职工信息表：
职工工号：1, 职工姓名：2, 职工工资：1111
请按任意键继续. . .
```

2 (10分)

从键盘输入一个字符串，要求将该字符串的内容按输入的相反顺序组织到一个不带表头结点的单链表中。假设输入的字符串为“abcdefg”，则组织到链表中的内容为“gfedcba”

```
1 #include <iostream>#include <cstring>
2 using namespace std;
3
4 // 定义单链表结点结构
5 struct Node {
6     char data;          // 存储字符
7     Node *next;         // 指向下一个结点的指针
8 };
9
10 // 将字符按反向顺序插入到链表中
11 void insertReverse(Node *&head, char ch) {
12     Node *newNode = new Node; // 创建新的结点
13     newNode->data = ch;
14     newNode->next = head;      // 新结点的 next 指向当前链表的头结点
15     head = newNode;           // 更新头结点为新结点
16 }
17
18 // 输出链表内容
19 void printList(Node *head) {
20     Node *temp = head;
21     while (temp != nullptr) {
22         cout << temp->data;
```

```

23         temp = temp->next;
24     }
25     cout << endl;
26 }
27
28 // 主函数
29 int main() {
30     char str[100]; // 用于存储输入的字符串
31     Node *head = nullptr; // 初始化链表为空
32
33     // 从键盘输入字符串
34     cout << "请输入一个字符串: ";
35     cin.getline(str, 100); // 读取字符串, 最多100个字符
36
37     // 将字符串的每个字符按反向顺序插入到链表中
38     for (int i = 0; str[i] != '\0'; i++) {
39         insertReverse(head, str[i]);
40     }
41
42     // 输出链表中的内容
43     cout << "按反向顺序组织的链表内容: ";
44     printList(head);
45
46     system("pause");
47     return 0;
48 }

```

代码解释:

1. **Node 结构体**: 定义了一个单链表结点, 包含一个字符 `data` 和一个指向下一个结点的指针 `next`。
2. **insertReverse 函数**: 将一个字符按反向顺序插入到链表中。每次插入时, 将新字符放在链表的前面 (即更新头结点)。这样最终得到的链表就是输入字符串的反向。
3. **printList 函数**: 遍历链表并打印每个结点的数据。
4. **main 函数**: 首先从键盘输入一个字符串, 并逐个字符将其按反向顺序插入到链表中。最后, 打印出链表的内容, 即反向顺序的字符串。

D:\2024\课内\C++\第十一章\11.2\Debug\11.2.exe

```

请输入一个字符串: dsbbiou
按反向顺序组织的链表内容: uoibbsd
请按任意键继续. . .

```