****



**程序设计综合实践课程报告**

**简单数据结构实验**

**学生姓名**

**学院名称 智能与计算学部**

**专 业 工科试验班(智能与计算类)**

**学 号**

# 1. 链表合并

## 1.1题目分析

**先将两个链表合并再用sort排序即可。**

## 1.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <algorithm>  using namespace std;    int main()  {      int T, i;      cin >> T;      while (T--) {          int m, n, len;          cin >> m >> n;          len = m + n;          int linklist[1000];          for (i = 0; i < len; i++)              cin >> linklist[i];          sort(linklist, linklist + len);          cout << linklist[0];          for (i = 1; i < len; i++)              cout << " " << linklist[i];          cout << endl;      }      return 0;  } |
|  |

# 2. 士兵队列训练问题

## 2.1题目分析

**可以用队列模拟，在需要的时候把某个元素舍去，而且不打乱原来的顺序。**

## 2.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  using namespace std;    int main()  {      int queue[5000];      int n;      cin >> n;      while (n--) {          int num, i, j, mod, \_num, flag = 0;          cin >> num;          // init          for (i = 0; i < num; i++)              queue[i] = i + 1;          \_num = num;          while (\_num > 3) {              \_num = num;              mod = flag ? 3 : 2;                int p = 0;              for (i = 0; i < num; i++) {                  if (queue[i] == 0)                      continue;                  else if ((p + 1) % mod != 0) {                      p++;                      continue;                  }                  else {// (p + 1) % mod == 0                      queue[i] = 0;                      p++;                  }              }              for (i = 0; i < num; i++)                  if (queue[i] == 0)                      \_num--;              flag = !flag;          }          int res[4];          j = 0;          for (i = 0; i < num; i++)              if (queue[i] != 0)                  res[j++] = queue[i];          for (i = 0; i < \_num; i++)              cout << res[i] << " ";          cout << endl;      }      return 0;  } |
|  |

# 3. Rails

## 3.1题目分析

**按照给出的出栈顺序，一个一个和当前栈顶的元素比较，不相等就向当前栈压入一个元素，因为是按照顺序压栈的。所以总会找到一个当前给出的顺序所在位置值一样的栈顶元素，然后出栈。同时给出的元素顺序也后移一位，直到给出的顺序遍历完或者出错。**

## 3.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  using namespace std;    int main()  {      int a[1000], b[1000], i, j, k, n;      while (cin >> n, n) {          while (cin >> b[0], b[0]) {              for (i = 1; i < n; i++)                  cin >> b[i];              for (i = 1, j = 0, k = 0; i <= n && j < n; i++, k++) {                  a[k] = i;                  while (a[k] == b[j]) {                      k--;                      j++;                      if (k == -1)                          break;                  }              }              if (j == n)                  cout << "Yes" << endl;              else                  cout << "No" << endl;          }          cout << endl;      }      return 0;  } |

# 4. Josephus Problem

## 4.1题目分析

**通过学习约瑟夫环的数学优化方法得到的解决方法。**

## 4.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  using namespace std;    int main()  {      int T;      cin >> T;      for (int p = 0; p < T; p++) {          int i, j, n, k, a[200];          cin >> n >> k;          // init          for (i = 0; i < n; i++)              a[i] = i + 1;          i = 0;          while (n > 1) {              i = (i + k - 1) % n;              for (j = i; j < n - 1; j++)                  a[j] = a[j + 1];              n--;          }          cout << "Case " << p + 1 << ": " << a[0] << endl;      }      return 0;  } |
|  |

# 5. Tree Recovery

## 5.1题目分析

**将树还原，根据先序和中序的结果还原树的形状，然后再后续遍历出结果。**

## 5.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  string pre, mid;    void post(int root, int start, int end)  {      if (start > end)          return;      int i = start;      while (i < end && mid[i] != pre[root])          i++;      post(root + 1, start, i - 1);      post(root + 1 + i - start, i + 1, end);      cout << mid[i];  }    int main()  {      int len;      while (cin >> pre >> mid) {          len = pre.length();          post(0, 0, len - 1);          cout << endl;      }      return 0;  } |
|  |

# 6. 四则运算

## 6.1题目分析

**需要确立优先级；使用操作数栈和运算符栈。**

## 6.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <stack>  #include <string.h>  using namespace std;  int main()  {      int T;      cin >> T;      while (T--) {          stack<double> s\_num;          stack<char> s\_op;          double num;          char op;          cin >> num;          while (op = getchar()) {              if (op == 13 || op == 10) {                  s\_num.push(num);                  break;              }              if (op == '+' || op == '-') {                  s\_num.push(num);                  s\_op.push(op);                  cin >> num;              }              else if (op == '\*') {                  double c;                  cin >> c;                  num = num \* c;              }              else if (op == '/') {                  double c;                  cin >> c;                  num = num / c;              }          }          while (!s\_op.empty()) {              double x = s\_num.top();              s\_num.pop();              double y = s\_num.top();              s\_num.pop();              char z = s\_op.top();              s\_op.pop();              double i;              if (z == '+')                  i = x + y;              else                  i = x - y;              s\_num.push(i);          }          double j = s\_num.top();          cout << j << endl;      }      return 0;  } |
|  |

# 7. 愚人节的礼物

## 7.1题目分析

**数字符左边有多少“（”和“）”即可，当遇到字符break。**

## 7.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  using namespace std;    int main()  {      char str[1005];      int i, numL, numR;      while (cin >> str) {          numL = numR = i = 0;          while (str[i] != 'B') {              if (str[i] == '(')                  numL++;              else                  numR++;              i++;          }          cout << numL - numR << endl;      }        return 0;  } |
|  |

# 8. Web Navigation

## 8.1题目分析

**本质上是利用栈来解决问题，再加上一些功能函数的书写。**

## 8.2 题目代码（带注释）

|  |
| --- |
| #include <iostream>  #include <stack>  using namespace std;    void test()  {      stack<string> for\_ward, back\_ward;      string temp, url, current;      current = "<http://www.acm.org/>";      while (cin >> temp) {          if (temp == "QUIT")              break;          if (temp == "VISIT") {              cin >> url;              back\_ward.push(current);              current = url;              cout << current << endl;              while (!for\_ward.empty())                  for\_ward.pop();              continue;          }          if (temp == "BACK") {              if (back\_ward.empty())                  cout << "Ignored" << endl;              else {                  for\_ward.push(current);                  current = back\_ward.top();                  back\_ward.pop();                  cout << current << endl;              }              continue;          }          if (temp == "FORWARD") {              if (for\_ward.empty())                  cout << "Ignored" << endl;              else {                  back\_ward.push(current);                  current = for\_ward.top();                  for\_ward.pop();                  cout << current << endl;              }              continue;          }      }  }  int main()  {      test();      return 0;  } |
|  |