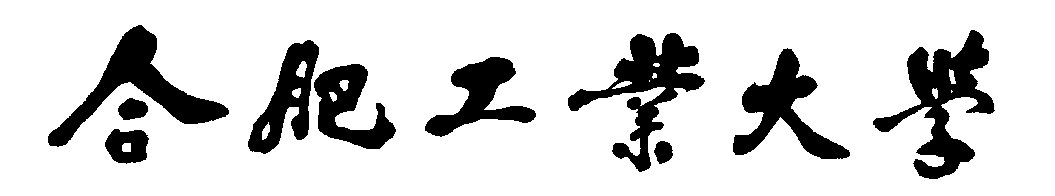
****

计算机与信息学院

数据结构实验报告

|  |  |
| --- | --- |
| 专 业 班 级 | 计算机科学与技术21-1 |
| 学生姓名及学号 | 杨程锦 2021214710 |
| 课程教学班号 |  |
| 任 课 教 师 | 胡学刚 |
| 实验指导教师 | 李培培 |
| 实验地点 | C201 |
| 2021 ~2022 学年第 二 学期 | |

说 明

实验报告是关于实验教学内容、过程及效果的记录和总结，因此，应注意以下事项和要求：

1．每个实验单元在4页的篇幅内完成一份报告。“实验单元”指按照实验指导书规定的实验内容。若篇幅不够，可另附纸。

2、各实验的**预习部分的内容是进入实验室做实验的必要条件**，请按要求做好预习。

3．实验报告要求：**书写工整规范，语言表达清楚，数据和程序真实。理论联系实际，认真分析实验中出现的问题与现象，总结经验。**

4．参加实验的每位同学应独立完成实验报告的撰写，其中程序或相关的设计图纸也可以采用打印等方式粘贴到报告中。严禁抄袭或拷贝，否则，一经查实，按作弊论取，并取消理论课考试资格。

5．实验报告作为评定实验成绩的依据。

实验序号及名称：实验

实验时间∶ 年 月 日

|  |
| --- |
| 预习内容 |
| 一、实验目的和要求∶  （1）理解三种线性结构栈、队列与线性表的特点；  （2）熟练掌握基于顺序表与基于链表的两种存储结构及  有关算法的设计。  （3）根据具体问题的需要，设计出合理的线性结构，并  设计相关算法。 |
| 二、实验任务∶  **任务一：**  **设计计算器系统，能对包含四则运算及括号的表达式计算其结果。**  任务二：  **“约瑟夫环问题”：约瑟夫，是一个古犹太人，曾经在一次罗马叛乱中担任将军，后来战败，他和朋友及另外39个人躲在一口井里。被发现后，罗马人表示只要投降就不死，约瑟夫和朋友想投降，可是其他人坚决不同意。怎么办呢? 让41个人围成一个圆圈，从第一个人开始报数，数到3的那个人被旁边的人杀死。这样就可以避免自杀了，因为犹太人的信仰是禁止自杀的。 约瑟夫和朋友站在什么位置才保住了性命呢？根据上述要求，设计算法输出约瑟夫和朋友所在位置。**  任务三：  **按照顺序表或者链表的存储结构分别实现串的类，并设计算法比较链串*S1*和链串*S2*的大小，若*S1* < *S2*，返回-1；若*S1* ＝ *S2*，返回0；否则返回1。** |
| 三、实验准备方案，包括以下内容：  （硬件类实验：实验原理、实验线路、设计方案等）  （软件类实验：所采用的核心方法、框架或流程图及程序清单）  任务一：**设计计算器系统，能对包含四则运算及括号的表达式计算其结果。**  把字符串压入sign栈内，数字压入number栈，读取输入的字符串，读到‘）’时，计算完‘（’之间的‘）’的所有字符，当sign栈空时，number栈中最后一个数字就是运算结果。    **任务二：“约瑟夫环问题”：**  从首结点开始，往后遍历循环链表，每三个删除一个结点，直到链表内只有head和另外两个结点为止。    **任务三：比较链串*S1*和链串*S2*的大小**  从两个链串的首节点开始比较，直到节点对应的data不同，比较data，data大的链串大于data小的链串 |

|  |
| --- |
| 实验内容 |
| 一、实验用仪器、设备：  Legion R9000p，cLion |
| 二、实验内容与步骤（过程及数据记录）：  任务一：**设计计算器系统，能对包含四则运算及括号的表达式计算其结果。**  #include<stack>  #include<iostream>  #include<string>  using namespace std;  bool isnumber(char x)//扫描数字的函数  {  if (x >= '0' && x <= '9')  return true;  return false;  }  int priority(char x) {//判断运算符优先级的函数  if (x == '+' || x == '-')  return 0;  else if (x == '\*' || x == '/')  return 1;  else if (x == '(' || x == ')')  return -1;  else if (x == '#')  return -2;  }  int calculate(string s)//计算  {  stack<int> number;//用于寄存操作数或运算结果  stack<char> operate;//用于寄存运算符  char top;  int a, b;  for (int i = 0; i < s.size(); ++i)  {  if (isnumber(s[i]))  {  int Count = 0;  string temp;//temp用于存放数字字符串，即计算数  temp += s[i];  while (isnumber(s[++i]))  temp += s[i];  for (int j = 0; j < temp.size(); ++j)  {  Count = Count \* 10 + temp[j] - 48;//数字字符的ASCII码与数字自身相差48  }  number.push(Count);  temp.clear();  }//将字符数字转换成整型数字  if (!isnumber(s[i]))  {  if (operate.empty())  {  operate.push(s[i]);  }//入栈第一个符号'#'  else  {  top = operate.top();  if (priority(s[i]) > priority(top) || s[i] == '(')  {  operate.push(s[i]);  }//入栈高优先级的运算符  else {  while (priority(s[i]) <= priority(top))  {  if (top == '#' && s[i] == '#')  {  int answer;  operate.pop();  answer = number.top();  cout << "\n答案是：" << answer << endl;  number.pop();  return 0;  }//当运算符实现完全后，只剩下'#'  else if (top == '(' && s[i] == ')')  {  ++i;  }//当左右括号相遇时，跳过右括号，删除左括号  else  {  a = number.top();  number.pop();  b = number.top();  number.pop();  }  if (top == '+')  {  b += a;  number.push(b);  }  else if (top == '-')  {  b -= a;  number.push(b);  }  else if (top == '\*')  {  b \*= a;  number.push(b);  }  else if (top == '/')  {  b /= a;  number.push(b);  }  operate.pop();  top = operate.top();;//取前一个运算符，用于与现在扫描的运算符进行比较  }//将优先级高的运算符实现计算  operate.push(s[i]);//用于当top=='#'时，将最后一个运算符入栈  }  }  }//扫描运算符，并判断优先级，以及运算  }//主循环  }  int main()  {  string s;  cout << "输入一个用'#'开头和结尾的表达式：" << endl;  cin >> s;  calculate(s);  cin.get();  return 0;  }    **任务二：“约瑟夫环问题”：**  #include "iostream"  using namespace std;  class Node {  public:  int data;  Node\* next;  };  class List {  public:  List();  ~List() {};  void Init();//创建链表  void Show();//遍历  void Round();  private:  int count;  Node\* head;  };  #include "JosephRing.h"  List::List() {  head = NULL;  count = 0;  }  void List::Init() {  cout << "请输入元素： " << endl;  int cur = 1;  Node\* ptr = new Node();  head = ptr;  ptr->data = cur;  ptr->next = ptr;  cur++;  count++;  while (cur != 42) {  Node\* s = new Node();  s->data = cur;  s->next = ptr->next;  ptr->next = s;  ptr = s;  cur++;  count++;  }  cout << "创建成功" << endl;  }  void List::Show() {  Node\* s = head;  for (int i = 0; i < count; i++)  {  cout << s->data << " ";  s = s->next;  }  cout << endl;  }  void List::Round() {  Node\* p = head;  while (count > 2) {  p = p->next;  Node\* a = p->next;  if (a == head) {  head = head->next;  }  p->next = p->next->next;  free(a); count--;  p = p->next;  }  }  #include "JosephRing.h"  int main() {  List j;  j.Init();  j.Show();  j.Round();  j.Show();  }    **任务三：比较链串*S1*和链串*S2*的大小**  #include "iostream"  using namespace std;  class Node{  public:  Node \*next;  char data;  };  class skewer {  private:  Node \*head;  int length;  public:  skewer();  void travelSkewer();  void createSkewer(string s);  int compareSkewer(skewer s2);  };  #include "skewer.h"  skewer::skewer() {  head = new Node();  head->next = NULL;  this->length = 0;  }  void skewer::createSkewer(string s) {  int lengths1 = s.length();  Node \*sym = new Node();  sym = head;  for (int i = 0; i < lengths1; i++){  Node \*p = new Node();  p->data = s[i];  sym->next = p;  sym = p;  length++;  }  }  void skewer::travelSkewer() {  Node \*p = new Node();  p = head->next;  for (int i = 0; i < length; i++){  cout << p->data;  p = p->next;  }  }  int skewer::compareSkewer(skewer s2) {  Node \*sym1 = new Node();  Node \*sym2 = new Node();  sym1 = this->head->next;  sym2 = s2.head->next;  int minLength;  if (this->length >= s2.length){  minLength = s2.length;  } else {  minLength = this->length;  }  for (int i = 0; i < minLength; i++){  if (sym1->data == sym2->data){  if (i == minLength - 1){  return 0;  } else {  sym1 = sym1->next;  sym2 = sym2->next;  continue;  }  } else if (sym1->data > sym2->data){  return 1;  } else {  return -1;  }  }  }  #include "skewer.h"  int main() {  skewer s1, s2;  string a = "afakj4325";  string b = "afakj4325";  s1.createSkewer(a);  s2.createSkewer(b);  cout << s1.compareSkewer(s2);  } |

|  |
| --- |
|  |
| 三、实验结果分析、思考题解答∶  实验结果正常，用栈实现了计算器，循环链表实现了约瑟夫环，以及链串的实现和比较。 |
| 四、感想、体会、建议∶  计算器和链串于我而言难度较小，计算器的计算等级方面思考了很长时间。约瑟夫换我尝试使用了多种数据结构，最后确定有循环链表，在使用循环链表时未考虑head结点，导致了很多的逻辑错误。 |
| 实验成绩∶  指导教师签名：  年 月 日 |