闭卷编程考试

第一题：读取输入字符串存储在数组中，将每个字符串按照8个字符切割，如果不足8个字符则自动补全为0，存储在新的字符串数组中，最后字符串按照升序输出。

测试样例：

输入：2 123456789 abc

输出：12345678 90000000 abc00000

（通过率只有80%，不知道为啥）

import java.util.\*;

public class Main {

static List<String> list = new ArrayList<>();

public static boolean CompareString(String s1,String s2){

char[] m1 = s1.toCharArray();

char[] m2 = s2.toCharArray();

for(int i = 0;i<8;i++){

if(m1[i] >= m2[i]){

return true;

}

else{//小于

return false;

}

}

return false;

}

public static String addzero(String result,int len){

for(int i=0;i<len;i++){

result += "0";

}

return result;

}

public static void splitby8( String result){

while(result.length()>8){

//继续切割

int l = result.length();

list.add(result.substring(0,8));

result = result.substring(8,l);

}

if(result.length()>0){

list.add(addzero(result,8-result.length()));

}

}

public static void main(String[] args) {

//如何实现升序？

Scanner in = new Scanner(System.in);

int num = Integer.valueOf(in.next());

String[] s = new String[num];

// String[] sresult = new String[num\*13];

for(int i=0;i<num;i++){

s[i]=in.next();

int sl = s[i].length();

if(sl>8){//按照8个8个切割，直到没有8个

splitby8(s[i]);

}

if(sl<8){

//自动补全0

list.add(addzero(s[i],8-sl));

}

}

String[] sresult = new String[list.size()];

for(int i =0;i<list.size();i++){

sresult[i] = list.get(i);

}

//打印

for(int i=0;i<sresult.length;i++){

for(int j=i;j<sresult.length;j++){

if(CompareString(sresult[i],sresult[j]))

//大于则交换

{

String temp = sresult[j];

sresult[j] = sresult[i];

sresult[i] = temp;

}

}

}

for(int i=0;i<sresult.length;i++){

if(i==sresult.length -1){

System.out.println(sresult[i]);

break;

}

System.out.print(sresult[i]+" ");

}

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

第二题题目：输入字符串只包括字母，数字和括号（大括号，中括号和小括号），数字n后括号内的字符串需要重复n遍，括号内可能有数字和括号。数字与括号一一对应，不存在2a1(b)的形式。然后逆序输出字符串

（当时未通过，后来补充）

测试用例：

输入1：abc3(A)

输出1：AAAcba

输入2：a2{b6[3(ac)]mg}h

输出2：hgmcacacacacacacacacacacacabgmcacacacacacacacacacacacaba

import java.lang.reflect.Array;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class haha {

//逆序字符数组

public static char[] reverseChar( char[] ch1){

char[] ch2 = new char[ch1.length];

int i =0;

for(int j = ch1.length-1; j>=0;j--){

ch2[i++] = ch1[j];

}

return ch2;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

String sum = in.next();

char[] my = sum.toCharArray();

String result = "";

//得到正序，括号里面可以出现括号，肯定先处理里面括号

int i =0;

//如果有大括号

if(sum.contains("{")){

while(i<my.length){

//考虑大括号

//只考虑小括号

if(my[i]>='0'&& my[i]<='9'){//遇到number需要处理

String mynum = "";

int mk = i;

while(my[mk] != '{'){

mynum += my[mk++];

}

int num = Integer.valueOf(mynum);

int h = mk+1;

String temp = "";

while(my[h] != '}'){

temp += my[h];

h++;

}

for(int k=0;k<num;k++) {

result += temp;

}

i = h+1;

}

else if(my[i]>='a'&&my[i]<= 'z'){//a-z

result += my[i++];

}

else if(my[i]>= 'A'&&my[i]<='Z'){//A-Z

result += my[i++];

}

else if(my[i] != '{'&&my[i] != '}'){

result += my[i++];

}

else{

i++;

}

}

}

//如果有中括号

i = 0;

if(result.isEmpty()) {

result = sum;

}

if(sum.contains("[")) {

char[] my1 = result.toCharArray();

String result1 = "";

while (i < my1.length) {

if (my1[i] >= '0' && my1[i] <= '9') {//遇到number需要处理

String my1num = "";

int mk = i;

while(my1[mk] != '['){

my1num += my1[mk++];

}

int num = Integer.valueOf(my1num);

int h = mk+1;

String temp = "";

while (my1[h] != ']') {

temp += my1[h];

h++;

}

for (int k = 0; k <num; k++) {

result1 += temp;

}

i = h + 1;

} else if (my1[i] >= 'a' && my1[i] <= 'z') {//a-z

result1 += my1[i++];

} else if (my1[i] >= 'A' && my1[i] <= 'Z') {//A-Z

result1 += my1[i++];

} else if (my1[i] != '[' && my1[i] != ']') {

result1 += my1[i++];

}

else{

i++;

}

}

result = result1;

}

i = 0;

if(result.isEmpty()) {

result = sum;

}

char[] my2 = result.toCharArray();

String result2 = "";

//后处理小括号

//如何找到括号最近的数字

while(i<my2.length){

//考虑大括号

//只考虑小括号

//需要确保是小括号前的数字

//从括号后面往前推导出数字

if (my2[i] > '0' && my2[i] <= '9') {//遇到number需要处理

String my2num = "";

int mk = i;

while (my2[mk] != '(') {

my2num += my2[mk++];

}

int num = Integer.valueOf(my2num);

int h = mk+1;

String temp = "";

while (my2[h] != ')') {

temp += my2[h];

h++;

}

for (int k = 0; k < num; k++) {

result2 += temp;

}

i = h + 1;

}

else if(my2[i]>='a'&&my2[i]<= 'z'){//a-z

result2 += my2[i];

}

else if(my2[i]>= 'A'&&my2[i]<='Z'){//A-Z

result2 += my2[i];

}

else if(my2[i] != ')'&&my2[i] != '('){

result2 += my2[i];

}

i++;

}

//逆序输出

char[] ch2 = reverseChar(result2.toCharArray());

String s = new String(ch2);

System.out.println(s);

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

第三题：破解小王的加密算法（没有时间做了……）