```
1  /*
2   * File:   addLib.h
3   * Author: inf-coccodi-de
4   *
5   * Created on 31 ottobre 2008, 11.36
6   */
7
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <errno.h>
11 #include <string.h>
12
13 char* cpyStr(char*);
14 int str2int(char*);
15 int lenStr(char*);
16 void errore(char*,int);
17 char* appendOnStr(char*,char);
18 char* removeFromStr(char*,char);
19 void permutazioni(char*,char*);
20 void debug(char*,int);
21 char* joinStr(char*,char*);
22 int cmpStr(char*,char*);
23 void fillWithZeroes(char*,int);
24
25 void fillWithZeroes(char* in,int length)
26 {
27  int i;
28  for(i=0;i<length;i++)
29    *(in+i)='\0';
30 }
31
32
33 void debug(char* s,int n)
34 {
35   #ifdef DEBUG
36    printf("%s %d\n",s,n);
37    fflush(stdout);
38   #endif
39 };
40
41 char* cpyStr(char* s)
42 {
43   char *ret,*b;
44   ret=(char*)malloc(sizeof(char)*lenStr(s+1)); //alloca  un'area di memoria
       grande come la lunghezza  della stringa s+1 e falla puntare da ret
45   for(b=ret;*b=*s;s++,b++); //copia di cella in cella da b a s finoa che s non
       vale 0
46   return(ret);
47 };
48
49 int lenStr(char*s)
50 {
51     int ret;
52     for (ret=0;*(s+ret);ret++);
53     return ret;
54 }
```

```
55
56  int str2int(char* s)
57  {
58      int ret;
59      for(ret=0;*s;s++)
60          ret=ret*10+*s-'0'; //sottrai '0' (30 ascii) al valore di s per
              ottenere il valore del numero intero cossispondente
61      return(ret);        //successivamente sommalo al valore di ret che e'
          stato moltiplicato per 10. Inserisci il tutto nella variabile ret
62  }
63
64  char* appendOnStr(char* s,char c)
65  {
66      char* ret,*b;
67      ret=(char*)malloc(sizeof(char*)*(lenStr(s)+2));
68      for(b=ret;*b=*s;s++,b++);
69      *b=c;
70      b++;
71      *b='\0';
72      return(ret);
73  };
74
75  /*char* appendOnStr(char *s,char c){
76      char *ret, *b;
77      ret=(char*)malloc(sizeof(char)*lenStr(s)+2);
78      b=ret;
79      while(*s){
80          *b=*s;
81          b++;
82          s++;
83      }
84      *b=c;
85      b++;
86      *b='\0';
87      return(ret);}*/
88
89  char* removeFromStr(char* s,char c)
90  {
91      char* ret,*b;
92      ret=(char*)malloc(sizeof(char)*lenStr(s));
93      b=ret;
94      while(*s)
95      {
96          if(*s!=c)
97          {
98              *b=*s;
99              b++;
100         }
101         s++;
102     }
103     *b='\0';
104     return (ret);
105 };
106
107 void errore(char* s,int n)
108 {
```

```
109     printf("%s, %d\n",s,n);
110     printf("ERRORE %d(%s)\n",errno,strerror(errno)); //stampa il numero
          dell'errore e la stringa che lo descrive
111     exit(3);
112 };
113
114 void permutazioni(char* sx, char* dx) {
115     char* newDx,*newSx;
116     int i,lenDx;
117     lenDx=lenStr(dx);
118     if(lenDx==1)
119     {
120         printf("%s%s\n",sx,dx);
121     }
122     else
123     {
124         for(i=0;i<lenDx;i++)
125         {
126             newSx=appendOnStr(sx,*(dx+i));
127             newDx=removeFromStr(dx,*(dx+i));
128             permutazioni(newSx,newDx);
129             free(newSx);
130             free(newDx);
131         }
132     }
133 };
134
135 char* joinStr(char* s1,char* s2)
136 {
137     char *ret,*tmp;
138     int lens,lent,lenComp;
139     lens=lenStr(s1);
140     lent=lenStr(s2);
141     lenComp=lens+lent+1;
142     //printf ("joinstrPrima:%d %d %d\n\n\n",lens,lent,lenComp);
143     //fflush(stdout);
144
145     //printf ("joinstrPrima:%s %s \n",s1,s2);
146     ret= (char*)malloc (sizeof(char)*lenComp);
147     //printf ("joinstrMeta:%s\n",ret);
148     for ( tmp =ret ; *tmp = *s1; s1++, tmp++);
149     for ( ; *tmp = *s2; s2++, tmp++);
150     *(tmp)='\0';
151     //printf ("joinstrDopo:%s\n\n\n",ret);
152     //fflush(stdout);
153     return (ret);
154 }
155
156
157 int cmpStr(char* s1,char* s2)
158 {
159  if(*s1>*s2) return(-1);
160  if(*s1<*s2) return(1);
161  if(*s1) return(cmpStr(++s1,++s2));
162  else return(0);
163 }
```