```
1  /*-------------------------------------------------------------------
2  Copyright (c) 2014 Author: Jagadeesh Vasudevamurthy
3  Filename: util.cpp
4  -------------------------------------------------------------------*/
5  #include "util.h"
6
7  /*-------------------------------------------------------------
8  strcmp
9  -------------------------------------------------------------*/
10 int Strcmp(const char* s1, const char* s2){
11   for (; *s1 == *s2; ++s1, ++s2)
12   if (*s1 == 0) {
13     return 0;
14   }
15   return *(unsigned char *)s1 < *(unsigned char *)s2 ? -1 : 1;
16 }
17
18 /*-------------------------------------------------------------
19 strcpy
20 -------------------------------------------------------------*/
21 void Strcpy(char* s1, const char* s2){
22   int i = 0;
23   while (1) {
24     s1[i] = s2[i];
25     if (s2[i] == '\0') {
26       return;
27     }
28     i++;
29   }
30 }
31
32 /*-------------------------------------------------------------------
33 Print an integer
34 -------------------------------------------------------------------*/
35 void print_integer(const int& x){
36   cout << x << " ";
37 }
38
39 /*-------------------------------------------------------------------
40 Print an integer
41 -------------------------------------------------------------------*/
42 void print_integer(const int*& x){
43   cout << *x << " ";
44 }
45
46 /*-------------------------------------------------------------------
47 Print an integer
48 -------------------------------------------------------------------*/
49 void print_integer(int& x){
50   cout << x << " ";
51 }
52
53 /*-------------------------------------------------------------------
54 Print an integer
55 -------------------------------------------------------------------*/
56 void print_integer(int*& x){
57   cout << *x << " ";
58 }
59
60 /*-------------------------------------------------------------------
61 7 9 1
62 9 7 -1
63 -------------------------------------------------------------------*/
64 int int_ascending_order(const int& c1, const int& c2){
65   if (c1 == c2) {
66     return 0;
```

```
  }
  if (c1 < c2) {
    return 1;
  }
  return -1;
}

/*-----------------------------------------------------------------
pointer
-----------------------------------------------------------------*/
int int_ascending_order(int* const& c1, int* const& c2){
  return int_ascending_order(*c1, *c2);
}

/*-----------------------------------------------------------------
7 9 -1
9 7  1
-----------------------------------------------------------------*/
int int_descending_order(const int& c1, const int& c2){
  int x = int_ascending_order(c1, c2);
  return -x;
}

/*-----------------------------------------------------------------
pointer
-----------------------------------------------------------------*/
int int_descending_order(int* const& c1, int* const& c2){
  return int_descending_order(*c1, *c2);
}

/*-----------------------------------------------------------------
Delete a int object
-----------------------------------------------------------------*/
void delete_int(int*& c){
  delete(c);
}

/*-----------------------------------------------------------------
Delete a char * object
-----------------------------------------------------------------*/
void delete_charstar(char*& c){
  delete[] c;
}

/*-----------------------------------------------------------
char compare
-----------------------------------------------------------*/
int charcompare(const char& c1, const char& c2){
  if (c1 == c2) {
    return 0;
  }
  if (c1 > c2) {
    return 1;
  }
  return -1;
}

/*-----------------------------------------------------------
char print
-----------------------------------------------------------*/
void print_char(char& c){
  cout << c << " ";
}

/*-----------------------------------------------------------
string print
```

```
133 ------------------------------------------------------------*/
134 void print_string(char*& c){
135   cout << c << " ";
136 }
137
138 /*-------------------------------------------------------
139 Delete c which is allocated by new []
140 ------------------------------------------------------------*/
141 void free_string(char*& c){
142   delete[] c;
143 }
144
145 /*-------------------------------------------------------
146 henry  zoo  is in descending order:  -1
147 tom   idiot is in decending order:    1
148 ------------------------------------------------------------*/
149 int string_descending_order(char* const& c1, char* const& c2){
150   int x = strcmp(c1, c2);
151   return x;
152 }
153
154 /*-------------------------------------------------------
155 henry  zoo  is in ascending order:  -1   1
156 tom   idiot is in ascending order:   1  -1
157 ------------------------------------------------------------*/
158 int string_ascending_order(char* const& c1, char* const& c2){
159   int x = string_descending_order(c1, c2);
160   return -x;
161 }
162
163
```