Programming Design, Spring 2016
Homework 9 Solution

Solution provider: Chien-Yu Huang
Department of Information Management
National Taiwan University

Problem 1

(a) The modified program is

```
char a[100] = {0};
while(cin.getline(a, 100))
{
  int i = 0;
  int spaceCount = 0;
  int lastSpace= -2;  //the index of the previous space

  while(a[i] != '\0')
  {
    if(a[i] ==' '&& lastSpace!=i-1){
      lastSpace=i;
      spaceCount ++;
    }
    else if(a[i] ==' ') {
      lastSpace=i;
    }

    i++;
  }
  cout << spaceCount << "\n";
}
```

(b) We may modify the code into:

```
char a[100] = {0};
cin.getline(a,100);
char* p = strstr(a, "ntu");
while(p != nullptr)
{
  char temp = *(p + 3);
  strcpy (p, "NTU");
  *(p + 3) = temp;
  p = strstr (p, "ntu");
}
cout << a << "\n";
```

Alternatively, we may also use function `strncpy` to obtain the desired string.

```
char a[100];
cin.getline(a,100);

char* p = strstr(a, "ntu");
while(p != nullptr) {

  strncpy(p, "NTU", 3);
  p++;
  p = strstr(p, "ntu");
}
cout << a << "\n";
```

(c) `strcpy` is invoked 15 times.

(d) In the main function, we create four pointers to keep track of the four character arrays' addresses. As we call `sortName`, if in its `for` loop we find names to swap, instead of copying the whole

character-array to another in problem (c), we simply swap the pointers that point to the two character-arrays. For efficiency, it is better to avoid actual swapping of data whenever a data item is large, such as a string or an entire database record. This approach saves a lot of time, with the additional advantage that the data items remain available in the original order.

Problem 2

Please see the attached CPP file.

Problem 3

Please see the attached CPP file.