ICS 103, Term 083

Computer Programming in C

HW# 3 Solution
Due date: Sunday, August 23, 2009

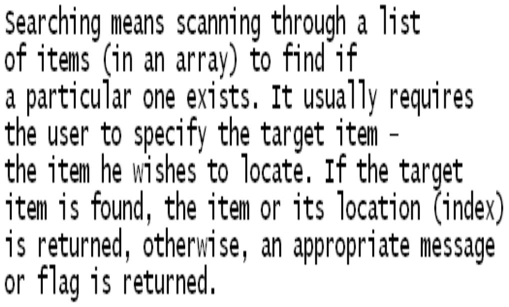
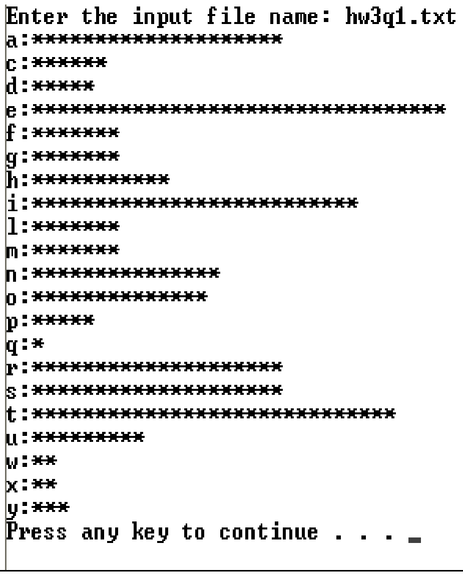
Q.1. You are required to write a C program to do the following:

(i) Ask the user to enter an input file name and read it.

(ii) Count the number of occurrences of each of the alphabetic characters in the input file i.e. ‘a’ to ‘z’ regardless whether the character is small letter or capital.

(iii) Write a function **histogram** that receives an input array and displays a histogram of ‘*’ according to each element count. If an array element count is 0, that element should not be displayed. Then, use this function to display a histogram of the counted alphabetic characters read from the file.

A sample execution of the program is shown below:

Input file: hw3.txt	*Histogram Display:*
 Searching means scanning through a list of items (in an array) to find if a particular one exists. It usually requires the user to specify the target item - the item he wishes to locate. If the target item is found, the item or its location (index) is returned, otherwise, an appropriate message or flag is returned.	 Enter the input file name: hw3q1.txt a:******************** c:****** d:***** e:********************************* f:******* g:******* h:*********** i:************************** l:******* m:******* n:*************** o:************** p:***** q:* r:******************** s:******************** t:***************************** u:********* w:** x:** y:*** Press any key to continue . . . _

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 26
void histogram(int x[], int size);

int main(void)
```

```
{

  FILE *inf;
  char infname[40], ch;
  int C[SIZE]={0}, i;


  printf("Enter the input file name: ");
  gets(infname);
  inf = fopen( infname, "r");
  if (inf == NULL){
     printf("Cannot open %s for reading \n", infname);
     system("pause");
     exit(1);
  }


  while ( fscanf(inf,"%c",&ch) != EOF ) {
        if ((ch>='a' && ch<='z')||(ch>='A' && ch<='Z')){
           ch = ch | 32; // convert chararcters to lower case
           C[(int)ch-97]++;
        }
  }

  histogram(C, SIZE);



  system("pause");

  return 0;
}

void histogram(int x[], int size){
   int i, j;
   for (i=0; i<size;i++)
      if (x[i] !=0){
         printf("%c:",i+97);
         for (j=0;j<x[i];j++)
             printf("*");
         printf("\n");
      }
}
```

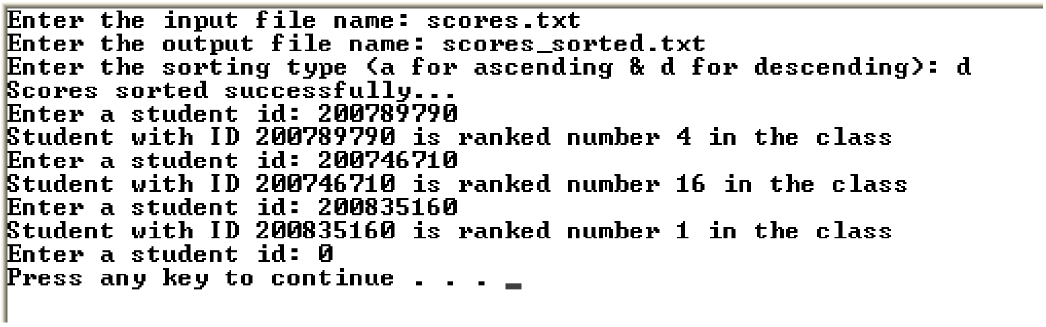
Q.2. You are required to write a C program to do the following:

(i) Ask the user to enter input file name and read from it students ID's along with their exam scores.

(ii) Ask the user to select the type of sorting required (**a** for ascending sorting of scores and **d** for descending). Then, write the sorted IDs and scores in an output file with its name selected by the user. Assume that the default sorting will be descending.

(iii) Ask the user to enter a student ID and display the rank of the student in the class. This process is repeated until the user enters 0 for the ID.

A sample execution of the program is shown below:



```
Enter the input file name: scores.txt
Enter the output file name: scores_sorted.txt
Enter the sorting type (a for ascending & d for descending): d
Scores sorted successfully...
Enter a student id: 200789790
Student with ID 200789790 is ranked number 4 in the class
Enter a student id: 200746710
Student with ID 200746710 is ranked number 16 in the class
Enter a student id: 200835160
Student with ID 200835160 is ranked number 1 in the class
Enter a student id: 0
Press any key to continue . . . _
```

scores.txt:	*scores_sorted.txt*

200835160 40.1
200793130 25.7
200791670 35.1
200789790 34.5
200781890 30.4
200781030 21.5
200773150 35.8
200771010 27.2
200758650 30.8
200746710 14.5
200745290 30.8
200721890 23.7
200719570 32.6
200717090 31.6
200640760 28.7
200614800 17.3
	Student ID Score
	---------- -----
	200835160 40.1
	200773150 35.8
	200791670 35.1
	200789790 34.5
	200719570 32.6
	200717090 31.6
	200758650 30.8
	200745290 30.8
	200781890 30.4
	200640760 28.7
	200771010 27.2
	200793130 25.7
	200721890 23.7
	200781030 21.5
	200614800 17.3
	200746710 14.5

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 100
```

```
void bubble_sort(int a[], double b[], int size, char stype);
void dswap(double *a, double *b);
void dswap(int *a, int *b);
int linear_search(int  a[],  int target, int size);

int main(void)
{

  FILE *inf, *outf;
  char infname[40], outfname[40], stype;
  int IDs[SIZE], count=0, i, id, index;
  double Scores[SIZE];

  printf("Enter the input file name: ");
  gets(infname);
  inf = fopen( infname, "r");
  if (inf == NULL){
     printf("Cannot open %s for reading \n", infname);
     system("pause");
     exit(1);
  }
  printf("Enter the output file name: ");
  gets(outfname);
  outf = fopen( outfname, "w");

  printf("Enter the sorting type (a for ascending & d for descending): ");
  scanf("%c", &stype);
  while ( fscanf(inf,"%d %lf",&IDs[count], &Scores[count]) != EOF ) {
        count++;
  }

// sorting scores & IDs according to scores

  bubble_sort(IDs, Scores, count, stype);

// printing IDs & scores into output file

  fprintf(outf,"Student ID \t Score\n");
  fprintf(outf,"---------- \t -----\n");
  for (i=0; i<count;i++)
      fprintf(outf, "%d \t %.1f\n",IDs[i], Scores[i]);

  printf("Scores sorted successfully...\n");

  do{
     printf("Enter a student id: ");
     scanf("%d",&id);
     if (id==0) break;
```

```
    index=linear_search(IDs, id, count);
    if (index != -1)
       printf("Student with ID %d is ranked number %d in the class\n", id, index+1);
     else
       printf("Student with ID %d is not found!!\n", id);

  } while (1);

  system("pause");

  return 0;
}

void dswap(double *a, double *b) {
    double temp = *a;
    *a = *b;
    *b = temp;
}
void iswap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
void bubble_sort(int a[], double b[], int size, char stype) {
    int i, pass = 1, swap_occurs;

    switch(stype){
    case 'a': case 'A':
        do{
          swap_occurs = 0;
          for(i = 1; i <= size - pass; i++)
               if (b[i - 1] > b[i]) {
                       dswap(&b[i-1], &b[i]);
                 iswap(&a[i-1], &a[i]);
                       swap_occurs = 1;
               }
          pass++;
        } while (swap_occurs && pass <= size-1);
        break;
    default:
        do{
          swap_occurs = 0;
          for(i = 1; i <= size - pass; i++)
               if (b[i - 1] < b[i]) {
                       dswap(&b[i-1], &b[i]);
                 iswap(&a[i-1], &a[i]);
                       swap_occurs = 1;
               }
          pass++;
```

```
            } while (swap_occurs && pass <= size-1);
        }

    }

    int linear_search(int a[],  int target, int size)
    {
        int i, found = 0,  where=-1;

        i = 0;
        while (!found && i < size) {
           if (a[i] == target){
                found = 1;
                where = i;
           } else
                ++i;
        }

        return where;
    }
```

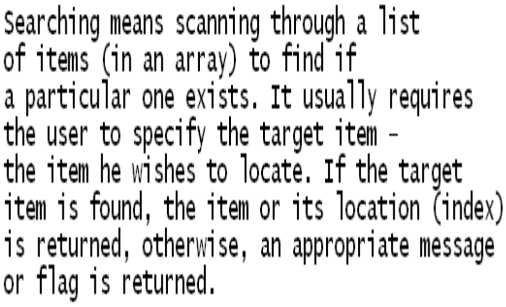
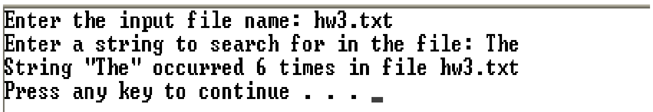
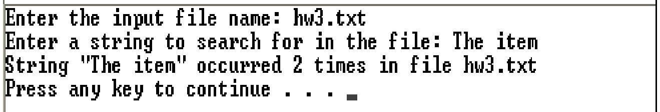
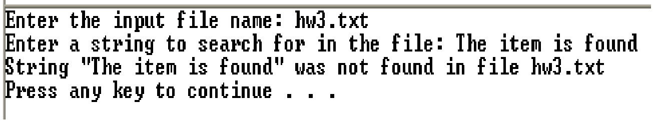
Q.3. You are required to write a C program to do the following:

(i) Ask the user to enter an input file name and read it.

(ii) Ask the user to enter a string and read it. Assume that the string is of maximum 80 characters and that the string can include spaces.

(iii) Count the number of occurrences of the string in the input file and display its count. Note that any matching string with the same letters regardless of being lower or upper case should be counted. If the string is not found, a message is printed indicating that the string was not found in the file.

A sample execution of the program is shown below:

Input file: hw3.txt	*Histogram Display:*
 Searching means scanning through a list of items (in an array) to find if a particular one exists. It usually requires the user to specify the target item - the item he wishes to locate. If the target item is found, the item or its location (index) is returned, otherwise, an appropriate message or flag is returned.	 Enter the input file name: hw3.txt Enter a string to search for in the file: The String "The" occurred 6 times in file hw3.txt Press any key to continue . . .  Enter the input file name: hw3.txt Enter a string to search for in the file: The item String "The item" occurred 2 times in file hw3.txt Press any key to continue . . .  Enter the input file name: hw3.txt Enter a string to search for in the file: The item is found String "The item is found" was not found in file hw3.txt Press any key to continue . . .

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void stolower (char str[], int size);

int main(void)
{

  FILE *inf;
  char infname[40], ostr[81], str[81], ch;
  int i, count=0;


  printf("Enter the input file name: ");
  gets(infname);
  inf = fopen( infname, "r");
  if (inf == NULL){
     printf("Cannot open %s for reading \n", infname);
     system("pause");
     exit(1);
  }

  printf("Enter a string to search for in the file: ");
  gets(str);
  strcpy(ostr,str);
  stolower(str, 80);
  i=0;
  while ( fscanf (inf,"%c",&ch) != EOF ) {

        if ((ch>='a' && ch<='z')||(ch>='A' && ch<='Z')){
           ch = ch | 32; // convert chararcters to lower case
           if (ch == str[i]) i++;
           else if (ch == str[0]) i=1;
           else i=0;
           if (i==strlen(str)) count++;
        }
  }
  if (count>0)
     printf("String \"%s\" occurred %d times in file %s\n",ostr, count, infname);
  else
      printf("String \"%s\" was not found in file %s\n", ostr, infname);

  system("pause");

  return 0;
}
```

```
void stolower (char str[], int size){

    for (int i=0; i<size; i++)
        if ((str[i]>='a' && str[i]<='z')||(str[i]>='A' && str[i]<='Z'))
            str[i] = str[i] | 32; // convert chararcters to lower case
}
```