DCP1203 HW10

12/5

Overview

- Announcements
- HW10

Announcements

Announcements

- CLASS-QUIZ next Thursday (12/14)
- Hand in HW10 before 12/10 24:00 if you want extra points for Lab-quiz 2.

HW10: Problems

1 Counting the Occurrences of a Character

- Write a program that inputs three lines of text and a search character and uses function strchr to determine the total occurrences of the character in three lines.

1 Counting the Occurrences of a Character

```
Enter three lines of text:
abc aa
123123 123
asd

Enter a search character: a

The total occurrences of 'a' in the text is 4

Process returned 0 (0x0)   execution time : 59.371 s
Press any key to continue.
```

2 Arrays of Pointers to Functions

- (Please refer to HW10_5.c)Complete the program to use a menu-driven interface.
- The program should offer the user four options as follows:
- (1)One restriction on using arrays of pointers to functions is that all the pointers must have the same type.

2 Arrays of Pointers to Functions

- (2)The pointers must be to functions of the same return type that receive arguments of the same type.
- (3)type and take the same parameters.
  Modify functions minimum and maximum to print the minimum or maximum value and return nothing.

2 Arrays of Pointers to Functions

- For option 3, modify function average to output the average for each student (not a specific student). Function average should return nothing and take the same parameters as printArray, minimum and maximum. Store the pointers to the four functions in array processGrades and use the choice made by the user as the subscript into the array for calling each function.

2 Arrays of Pointers to Functions

```
Enter a choice:
   0  Print the array of grades
   1  Find the minimum grade
   2  Find the maximum grade
   3  Print the average on all tests for each student
   4  End program
```

2 Arrays of Pointers to Functions

```
Enter a choice:
  0  Print the array of grades
  1  Find the minimum grade
  2  Find the maximum grade
  3  Print the average on all tests for each student
  4  End program
? 0

                  [ 0 ]  [ 1 ]  [ 2 ]  [ 3 ]
studentGrades[ 0 ] 77     68     86     73
studentGrades[ 1 ] 96     87     89     78
studentGrades[ 2 ] 70     90     86     81
```

2

```
Enter a choice:
  0  Print the array of grades
  1  Find the minimum grade
  2  Find the maximum grade
  3  Print the average on all tests for each student
  4  End program
? 1

The lowest grade is 68

Enter a choice:
  0  Print the array of grades
  1  Find the minimum grade
  2  Find the maximum grade
  3  Print the average on all tests for each student
  4  End program
? 2

The highest grade is 96
```

2 Arrays of Pointers to Functions

```
Enter a choice:
  0  Print the array of grades
  1  Find the minimum grade
  2  Find the maximum grade
  3  Print the average on all tests for each student
  4  End program
? 3

The average for student 1 is 76.0
The average for student 2 is 87.5
The average for student 3 is 81.8
```

3

- Hmmmmmm..... strings again :) Then it must be an easy task for you. Well . . . you are given with a string S of length not more than 100,000 (only `a'-`z' and `A'-`Z'). Then follows q (q < 1000) queries where each query contains a string T of maximum length 1,000 (also contains only `a'-`z' and `A'-`Z').

3

- You have to determine whether or not T is a sub-string of S.
- Note: use malloc
- Input: First line contains an integer k (k < 10) telling the number of test cases to follow. Each test case begins with S. It is followed by q. After this line there are q lines each of which has a string T as described before.
- Output: For each query print `y' if it is a substring of S or `n' otherwise followed by a new line. See the sample output below.

3

Sample Input

```
2
abcdefghABCDEFGH
2
abc
abAB
xyz
1
xyz
```

Sample Output

```
y
n
y
```