

CGS3460 Fall 2007
Assignment 9

Assigned: 10/20, Sat

Due: 10/26, Fri

1. Write a menu driven program (see examples) that allows the user to choose among the following options:

1. **Palindrome**: A palindrome is a word that reads the same backward as forward. If the user selects this option, ask the user to input a string (character array) and report back if the entered string is a palindrome or not.
2. **x^a** : If the user selects this option, ask the user for the values of x (floating point number) and a (integer) and report back the value of x raised to the power a . Display your result till 6 decimal places.
3. **$x!$** : If the user selects this option, ask the user to enter the value for x and report back the value of x (positive integer) factorial. Display your result till 6 decimal places.
4. **No. of occurrences**: If the user selects this option, ask the user to enter a string and report back the number of occurrences of the first character of the string in the whole string (including the first one i.e. there will always be at least 1 occurrence).
5. **Binary search**: If the user selects this option, ask the user to enter a list of numbers (user terminates the list with a -1) and a number that user wants to search. Report back if the asked number was found in the list or not. The search must use the Binary search algorithm. You can find more details for Binary Search Algorithm at http://en.wikipedia.org/wiki/Binary_search.
6. **Exit**: If the user selects this option, your program must terminate.

You must implement each one of the above algorithms using **recursion**. Once a selected option has been completed, the program must redisplay the menu and ask again for the user's choice. This must continue till user explicitly chooses to exit the program by selecting option # 6. You don't have to handle the cases with invalid inputs. Save this program in a file called p1.c. Look at the following examples to see what messages are to be displayed.

Example 1:

Choose one of the following:

1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit

Enter your choice: 1

Enter a string (max 20 char): abba

This string is a palindrome

Choose one of the following:

1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit

Enter your choice: 2

```
Enter value for x:1.1
Enter value for a:2
x^a = 1.210000
Choose one of the following:
1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 3
Enter value for x:5
x! = 120
Choose one of the following:
1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 4
Enter a string (max 20 char): this is a test
t occurs 3 times in 'this is a test'
Choose one of the following:
1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 5
Enter integers, ending with -1.
12 3 4 2 3 34 5 34 -1
Enter the number to be searched: 3
Number found
Choose one of the following:
1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 6
```

Example 2:

```
Choose one of the following:
1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit
```

Enter your choice: 1
Enter a string (max 20 char): test
This string is not a palindrome
Choose one of the following:
1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 6

Example 3:

Choose one of the following:
1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 2
Enter value for x:2.344
Enter value for a:0
 $x^a = 1.000000$
Choose one of the following:
1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 6

Example 4:

Choose one of the following:
1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 4
Enter a string (max 20 char): zooooo
z occurs 1 times in 'zooooo'
Choose one of the following:
1. Palindrome
2. x^a
3. $x!$
4. Number of occurrence
5. Binary Search
6. Exit
Enter your choice: 6

Example 5:

Choose one of the following:

1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit

Enter your choice: 5

Enter integers, ending with -1.

1 2 3 2 1 3 4 52 -1

Enter the number to be searched: 100

Number not found

Choose one of the following:

1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit

Enter your choice: 5

Enter integers, ending with -1.

12 -1

Enter the number to be searched: 12

Number found

Choose one of the following:

1. Palindrome
2. x^a
3. x!
4. Number of occurrence
5. Binary Search
6. Exit

Enter your choice: 6

[Hint: If you want to, you can use `strlen()` function which returns the length of the string provided as its argument. To use this function you must include `string.h`. Also, `gets()` function can be used to obtain a string (character array) input easily. Both of these functions are demonstrated in following example.

Example:

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str[256];
    printf("Enter a string: ");
    gets(str);
    printf("Length of entered string is %d", strlen(str));
}
```

On Execution:

```
Enter a string: this is a test  
Length of entered string is 14]
```

Instructions for submitting Assignment 9

1. First compress your program files into a single file. At the terminal command prompt (Putty incase you are working from Windows) type **tar cvf assign9.tar p1.c**. This should create the compressed file assign9.tar
2. If you are working from Windows platform using Putty and WinSCP, you can transfer the tar file back to your Windows machine using WinSCP. It is reverse of the process you probably used to transfer your program(s) from your local machine to CISE machine.
3. Upload the assign9.tar to CourseWorX at the course webpage.