DAY 11-DAILY ASSIGNMENTS

ANSU MARIUM SHIBU

20-11-2024

1. Problem 1: Palindrome Checker

Problem Statement:

Write a C program to check if a given string is a palindrome. A string is considered a palindrome if it reads the same backward as forward, ignoring case and non-alphanumeric characters. Use functions like strlen(), tolower(), and isalpha().

Example:

Input: "A man, a plan, a canal, Panama"

Output: "Palindrome"

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int isAlphabetic(char c);
int isPalindrome(char *str);

int main() {
    char str[1000];

    printf("Enter a string: ");
    scanf("%[^\n]%*c", str);

    if (isPalindrome(str)) {
        printf("Palindrome\n");
    } else {
        printf("Not a palindrome\n");
    }

    return 0;
}
int isAlphabetic(char c) {
    return (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z');
}

int isPalindrome(char *str) {
    int left = 0, right = strlen(str) - 1;

    while (left < right) {
```

```c
int isPalindrome(char *str) {
    int left = 0, right = strlen(str) - 1;

    while (left < right) {
        while (left < right && !isAlphabetic(str[left])) {
            left++;
        }
        while (left < right && !isAlphabetic(str[right])) {
            right--;
        }
        if (tolower(str[left]) != tolower(str[right])) {
            return 0;
        }
        left++;
        right--;
    }

    return 1;
}
```

```
PS D:\c progrms coding> ./a
Enter a string: A man, a plan, a canal, Panama
Palindrome
PS D:\c progrms coding> gcc strassi4.c
```

2. Problem 4: Reverse Words in a Sentence

Problem Statement:

Write a program to reverse the words in a given sentence. Use strtok() to extract words and strcat() to rebuild the reversed string.

Example:

Input: "The quick brown fox"

Output: "fox brown quick The"

```c
C strassi4.c > main()
#include <stdio.h>
#include <string.h>

int main() {
    char str[100], revstr[100] = "";
    printf("Enter sentence: ");
    scanf("%[^\n]%*c", str);

    char *token = strtok(str, " ");

    while (token != NULL) {
        if (strlen(revstr) > 0) {
            char temp[100];
            strcpy(temp, revstr);
            strcpy(revstr, token);
            strcat(revstr, " ");
            strcat(revstr, temp);
        } else {
            strcpy(revstr, token);
        }
        token = strtok(NULL, " ");
    }

    printf("Reversed sentence: %s\n", revstr);
    return 0;
}
```

```
PS D:\c progrms coding> gcc strassi4.c
PS D:\c progrms coding> ./a
Enter sentence: ansu marium shbiu
Reversed sentence: shbiu marium ansu
```

3. Problem 5: Longest Repeating Substring

Problem Statement:

Write a program to find the longest substring that appears more than once in a given string. Use strncpy() to extract substrings and strcmp() to compare them.

Example:

Input: "banana"

Output: "ana"

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

int main() {
    char str[100], longest[100] = "";
    int n, maxLength = 0;

    printf("Enter a string: ");
    scanf("%s", str);
    n = strlen(str);

    for (int len = 1; len < n; len++) {
        for (int i = 0; i <= n - len; i++) {
            char substring[100];
            strncpy(substring, str + i, len);
            substring[len] = '\0';

            for (int j = i + 1; j <= n - len; j++) {
                char compare[100];
                strncpy(compare, str + j, len);
                compare[len] = '\0';

                if (strcmp(substring, compare) == 0 && len > maxLength) {
                    maxLength = len;
```

```c
    for (int len = 1; len < n; len++) {
        for (int i = 0; i <= n - len; i++) {
            for (int j = i + 1; j <= n - len; j++) {
                strncpy(compare, str + j, len);
                compare[len] = '\0';

                if (strcmp(substring, compare) == 0 && len > maxLength) {
                    maxLength = len;
                    strcpy(longest, substring);
                }
            }
        }
    }

    if (maxLength > 0) {
        printf("Longest repeating substring: %s\n", longest);
    } else {
        printf("No repeating substring found.\n");
    }

    return 0;
}
```

```
Longest repeating substring: ann
Enter a string: annnsuuuuannsndjdd
Longest repeating substring: ann
```

4. Problem 2: Word Frequency Counter

Problem Statement:

Write a program to count the frequency of each word in a given string. Use strtok() to tokenize the string and strcmp() to compare words. Ignore case differences.

Example:

Input: "This is a test. This test is simple."

Output:

Word: This, Frequency: 2

Word: is, Frequency: 2

Word: a, Frequency: 1

Word: test, Frequency: 2

Word: simple, Frequency: 1

assi2.c > ...

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main() {
    char str[500];
    char *token;
    char words[100][50];
    int freq[100] = {0};
    int wordCount = 0;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    token = strtok(str, " .");
    while (token != NULL) {
        for (int i = 0; token[i]; i++) {
            token[i] = tolower(token[i]);
        }

        int found = 0;
        for (int i = 0; i < wordCount; i++) {
            if (strcmp(words[i], token) == 0) {
                freq[i]++;
                found = 1;
                break;
            }
        }
```

```
                break;
            }
        }

        if (!found) {
            strcpy(words[wordCount], token);
            freq[wordCount] = 1;
            wordCount++;
        }

        token = strtok(NULL, " .");
    }

    for (int i = 0; i < wordCount; i++) {
        printf("Word: %s, Frequency: %d\n", words[i], freq[i]);
    }

    return 0;
}

```

```
PS D:\c progrms coding> ./a
Enter a string: ansu amrium shibu
Word: ansu, Frequency: 1
Word: amrium, Frequency: 1
Word: ansu, Frequency: 1
Word: amrium, Frequency: 1
Word: shibu
, Frequency: 1
PS D:\c progrms coding>
```

5. Problem 3: Find and Replace

Problem Statement:

Create a program that replaces all occurrences of a target substring with another substring in a given string. Use strstr() to locate the target substring and strcpy() or strncpy() for modifications.

Example:

Input:

String: "hello world, hello everyone"

Target: "hello"

Replace with: "hi"

Output: "hi world, hi everyone"

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

int main() {
    char str[1000], target[100], replacement[100];
    int i, j, k, targetLen, replacementLen;


    printf("Enter the string: ");
    scanf("%[^\n]", str);

    printf("Enter the target substring: ");
    scanf(" %[^\n]", target);

    printf("Enter the replacement substring: ");
    scanf(" %[^\n]", replacement);

    targetLen = strlen(target);
    replacementLen = strlen(replacement);


    for (i = 0, k = 0; str[i] != '\0'; i++) {

        if (strstr(&str[i], target) == &str[i]) {
            for (j = 0; j < replacementLen; j++) {
                str[k++] = replacement[j];
            }
            i += targetLen - 1;
        } else {
```

```c
printf("Enter the replacement substring: ");
scanf(" %[^\n]", replacement);

targetLen = strlen(target);
replacementLen = strlen(replacement);


for (i = 0, k = 0; str[i] != '\0'; i++) {

    if (strstr(&str[i], target) == &str[i]) {
        for (j = 0; j < replacementLen; j++) {
            str[k++] = replacement[j];
        }
        i += targetLen - 1;
    } else {
        str[k++] = str[i];
    }
}
str[k] = '\0';


printf("Modified string: %s\n", str);

return 0;
```

```
PS D:\c progrms coding> gcc strassi3.c
PS D:\c progrms coding> ./a
Enter the string: hlo hw r u
Enter the target substring: hlo
Enter the replacement substring: hi
Modified string: hi hw r u
PS D:\c progrms coding>
```

6. Declare string A and declare string B and copy value B to A and use array and pointer notation to print the string also use choice :when p enters it to print pointer notation and a enter prints array notation

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

void copystrarr(char to[], char from[]);
void copystrptr(char *to, char *from);

int main() {
    char a[20];
    char b[50] = "ansu";
    char choice;

    printf("Enter 'a' for array notation or 'p' for pointer notation: ");
    scanf(" %c", &choice);

    if (choice == 'a') {
        copystrarr(a, b);
        printf("Copied string using array notation: %s\n", a);
    }
    else if (choice == 'p') {
        copystrptr(a, b);
        printf("Copied string using pointer notation: %s\n", a);
    }
    else {
        printf("Invalid choice. Please enter 'a' or 'p'.\n");
    }

    return 0;
}
```

```c
    if (choice == 'a') {
    else if (choice == 'p') {
        copystrptr(a, b);
        printf("Copied string using pointer notation: %s\n", a);
    }
    else {
        printf("Invalid choice. Please enter 'a' or 'p'.\n");
    }

    return 0;
}

void copystrarr(char to[], char from[]) {
    int i;
    for (i = 0; from[i] != '\0'; ++i) {
        to[i] = from[i];
    }
    to[i] = '\0';
}

void copystrptr(char *to, char *from) {
    for (; *from != '\0'; ++from, ++to) {
        *to = *from;
    }
    *to = '\0';
}
```

```
PS D:\c progrms coding> gcc strcnv.c
PS D:\c progrms coding> ./a
Enter 'a' for array notation or 'p' for pointer notation: a
Copied string using array notation: ansu
PS D:\c progrms coding> gcc strcnv.c
PS D:\c progrms coding> ./a
Enter 'a' for array notation or 'p' for pointer notation: p
Copied string using pointer notation: ansu
PS D:\c progrms coding>
```