

Grundidee

Anfangen habe ich die Übung mit den verschiedenen Möglichkeiten der String Funktion. Das hat mir einen relativ kompakten Code ergeben, welcher aber auf eine Standardlibrary zugriff. Als Mehrwert habe ich mich entschlossen, das ganze Programm selber zu schreiben und somit die String Überprüfung selber vorzunehmen.

Der Code ist somit zwar etwas länger, jedoch konnte ich ihn so optimieren, dass er jetzt eine sehr schöne Grösse hat. Weiter habe ich das neue Interface für meine Praktikas so eingebaut, dass jetzt Auswahlen getroffen werden können.

main(args)

```
int main(args){
    char string[100];
    char* result;
    int option = 1;

    string = args;

    result = isPalindrome(string);

    if (result == 0) {
        printf("\'%s\' is not a palindrome string.\n", string);
    } else {
        printf("\'%s\' is a palindrome string.\n", string);
    }

    (void) printf("Would you like to continue? [0/1]?\n");
    (void) scanf("%d", &option);
    fflush(stdin);

}
```

main() Zusatz / Interface

Die momentane Zusatzfunktion ist im Moment nicht mehr implementiert. Jedoch ist es relativ simpel die Parameterübergabe für die Konsole zu implementieren. Und zwar sollen dazu Argumente an die main(arg) Funktion übergeben werden. Ich habe diese Funktionalität zugunsten des Interfaces wieder herausgenommen.

Interface

```
(void) printf("#####\n");
(void) printf("#          PROG C          #\n");
(void) printf("#          bachmste          #\n");
(void) printf("#          Palindrom          #\n");
(void) printf("#####\n\n");

while(option) {
    (void) printf("Please enter a String:\n");
    gets(string);

    result = isPalindrome(string);
if - Teil...

    (void) printf("Would you like to continue? [0/1]?\n");
    (void) scanf("%d", &option);
    fflush(stdin);
}
}
```

isPalindrome Funktion char*

Die eigentliche Hauptfunktion ist bei mir die isPalindrome Funktion. Diese gibt einen Pointer auf ein char* zurück, falls die Eingabe ein Palindrom ist. Ist diese kein Palindrom wird ein Null Pointer zurückgegeben.

HeaderSnippetlet

```
char* isPalindrome(char *string){
    int check, length;
    char *reverse;

    length = strlen(string);
    reverse = (char*)malloc((size_t) (length+1));

    copyString(reverse, string);
    reverseString(reverse);

    check = compareString(string, reverse);

    free(reverse);

    if(check == 0){
        return string;
    } else {
        return 0;
    }
}
```

Output der Funktion

Den Output habe ich so angepasst, dass jeweils gesagt wird, wieviele Elemente im gesamten im Array gespeichert sind. Ausserdem gibt es den ganzen String aus. Um mehrere Strings auszugeben, müsste ein zweidimensionales Array benutzt werden.

String compare und copy

```
int compareString(char *first, char *second){
    while(*first == *second){
        if(*first == '\0' || *second == '\0'){
            break;
        }
        first++;
        second++;
    }
    if(*first == '\0' && *second == '\0'){
        return 0;
    } else {
        return -1;
    }
}

void copyString(char *target, char *source){
    while(*source){
        *target = *source;
        source++;
        target++;
    }

    *target = '\0';
}
```