

CSE 102 : Structured Programming Lab

Lab Manual - 8

Objectives

- Usage of Strings

Problem 1: Basic String input and output

We will use `gets()` function to take input into the string from the user and `puts()` function to print the string as output. Write a program that takes the name, address and email address of the user and then prints them on the screen.

```
#include<stdio.h>
int main(){
    char name[50];
    char address[200];
    char email[50];

    printf("Put your name here: ");
    gets(name);
    printf("Put your address here: ");
    gets(address);
    printf("Put your email here: ");
    gets(email);

    printf("\nYour name is: ");
    puts(name);
    printf("Your address is: ");
    puts(address);
    printf("Your email is: ");
    puts(email);
}
```

Problem 2: Using ASCII values

Write a program that a string as input from the user. The program then finds out if there are any letters other than lower-case alphabets in the string.

```
#include<stdio.h>
int main(){
    int i=0, flag=0;
    char str[20];
    printf("Input string: ");
    gets(str);
    for(i=0; str[i] != '\0'; i++){
        if(str[i] < 97 || str[i] > 122){
            if(str[i] != 32){
                flag=1;
                break;
            }
        }
    }
    if(flag != 0){
        printf("There are characters other than lower-case alphabets!");
    }
    else{
        printf("There are only lower-case alphabets!");
    }
    return 0;
}
```

Problem 3: Using functions from the String library

Write a program that takes a string from a user as input. The program checks whether the string is a palindrome or not. You can use built-in library functions. A palindrome is a word that reads the same backwards as forwards. **RACECAR** is a palindrome.

```
#include<stdio.h>
#include<string.h>
int main(){
    int result;
    char s1[100], s2[100];
    printf("Input string to check: ");
    gets(s1);

    strcpy(s2, s1);
    strrev(s2); //strrev() reverses the string
    result = strcmp(s2, s1);

    if(result){
        printf("%s is not a palindrome!", s1);
    }
    else{
        printf("%s is a palindrome!", s1);
    }
    return 0;
}
```

The `strrev()` function reverses a string. It takes the string as input parameters and reverses the original string. It does not produce any output string but rather modifies the existing string to reverse it. Hence, it is a **in-place** function.

Task 1 (Try yourself):

Write a program that takes a string as input from the user. The program counts the total number of Upper-case characters, lower-case characters, and special characters in the string and outputs their numbers.

Sample input: NinJ@s are C00l

Sample output:

Upper-case letters: 3

Lower-case letters: 7

Special letters: 3

Task 2 (Try yourself)

Write a program that takes a string as input from the user and checks whether it is a palindrome or not. You are not allowed to use any library function. Try to come up with an algorithm that can give us the result in the shortest amount of time.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 1: ASCII Table