Characters Operation in C

1. **WAP in C to enter a character and check whether the entered character is vowel or consonant.**

```
#include <stdio.h>
#include <conio.h>

int main()
{
	char ch, chl;

	printf("Enter a character: ");
	scanf(" %c", &ch);

	// Convert character to lowercase or uppercase to simplify checking
	chl = tolower(ch);

	// NOTE: In C, single character should be quoted by single quote, not the double quote.
	// NOTE: In C, you only double quote string literals(arrays of characters.)

	if(chl == 'a' || chl == 'e' || chl == 'i' || chl == 'o' || chl == 'u')

		printf("%c is a vowel.\n", ch);

	else if (chl >= 'a' && chl <= 'z')

		printf("%c is a consonant.\n", ch);
	else
		printf("%c is not an alphabet.\n", ch);

	getch();

	return 0;
}
```

2. **Best way & multiple ways to hold string in C program.**

```
#include <stdio.h>

int main()
{
	char name[100];

	printf("Enter your full name: ");

	// // Alternative; It is used to read string input (i.e multiple words)
	//	gets(name);


	// Alternative way to read string input
	// fgets(name, sizeof(name), stdin);

	// // It is used to read single word input (can only hold firstName)
	//	scanf("%s", name);

	printf("\n%s\n", name);

	// // It is used to output string
	//	puts(name);

	return 0;
}
```

3. **Write a C program that takes string as input and counts the number of vowels and consonants in it.**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>            // <string.h> is for strlen() function

int main() {

        char str[100];
        int i, length, vowels = 0, consonants = 0;

        printf("Enter a string: ");
        gets(str);

        length = strlen(str);

        for (i = 0; i < length; i++) {

                char ch = tolower(str[i]);

                if(ch >= 'a' && ch <= 'z')
                {
                        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
                                vowels++;
                        else
                                consonants++;
                }

        }

        printf("\n Vowels count: %d \n", vowels);
        printf("\n Consonants count: %d \n", consonants);

        getch();

        return 0;
}
```

4. **WAP in C to ask the user to input his/her full name & display the name.**

```
#include <stdio.h>
#include <conio.h>

void main() {

        char fullName[100];

        printf("Enter fullName: ");
        gets(fullName);

        printf("\n %s \n", fullName);

    getch();

}
```

5. **WAP in c to enter a string and display the length of the string.**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>          // string.h is included for strlen() function.

int main(){

        int length;
        char str[100];

        printf("Enter a string: ");
        gets(str);

        length = strlen(str);

        printf("\nLength of string is %d.\n", length);

        getch();
        return 0;
}
```

6. **Properly handle single character input.**

```
#include <stdio.h>
#include <conio.h>

void main() {

        // Single character variable
        char ch, c;

        clrscr();

        printf("Enter a character: ");
        scanf(" %c", &ch);
        printf("\nEntered character is: %c\n", ch);

        printf("\nAgain,enter a character: ");
        scanf(" %c", &c); // should add a space before %c, if there was previous scanf() statement
        printf("\nEntered character is: %c\n", c);

        getch();
}
```

7. **WAP in C to enter a string and display its reverse.**

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


int main(){

        char a[100];

        clrscr();

        printf("Enter a string: ");
        gets(a);

        printf("\nReverse is: %s \n", strrev(a));

        getch();
        return 0;
}
```

8. **Write a C program to check if a string is palindrome or not.**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>                    // to use strcpy()

void main()
{
        int compare;
        char a[100], b[100];

        clrscr();

        printf("Enter a string: ");
        gets(a);


        // converting string to uppercase or lowercase for correct(flawless) comparison
        strupr(a);
        // Or, strlwr(a);


        // SYNTAX of strcpy() ==> strcpy(destination_string, source_string);
        strcpy(b, a);


        // Reversing string a using strrev()
        strrev(a);


        // Comparing string a & b
        // strcmp() gives 0 if found the same string but gives -1 if found the different strings.
        compare = strcmp(a, b);
        printf("Comparison = %d", compare);


        if (compare == 0)
                printf("\n The string %s is palindrome. \n", b);
        else
                printf("\n The string %s is not palindrome. \n", b);


        getch();
}
```

9. Asdfsadf

10.