STRING exercises

Ex1. Write a program that accepts a string `str` from the user and displays `str` on the screen.

For example, if you enter the following string:

```
Pham Ngoc Tho
```

The output should be:

```
Pham Ngoc Tho
```

Ex2. Write a program that accepts a name and address of a person from the keyboard and displays information on the screen as below:

```
Name: {P1}
Address: {P2}
```

Where `{P1}` is the name and `{P2}` is the address of that person.

For example, if you enter the following values:

```
Tho NamDinh
```

the code will produce the following result:

```
Name: Tho
Address: NamDinh
```

Note: do not enter the name or address having spaces

Ex3. Write a program that accepts a string `s` and an integer `k` from the user and displays the `k`'th character in string `s`.

For example, if `s = "fpt_university", k = 6`, enter the following values:

```
fpt_university
6
```

When the above code is compiled and executed, it produces the following result:

```
n
```

Because the 6-th character in `fpt_university` is `n`

Ex4. Write a program that accepts a string `str` from the user and displays the length of the given string on the screen.

For example, if you enter the following value:

```
Fpt University
```

The code will produce the following result:

```
14
```

Ex5. Given a string `s` and a character `c`. Write a program that accept these two variables from the user and prints the occurences of character `c` in `s`.

For example, if `s = "Codelearn", c = 'e'`, enter the following values:

```
Codelearn
e
```

When the code is compiled and executed, it produces the following result:

```
2
```

Because `'e'` appears 2 times in `"Codelearn"`

Ex6. Write a program that accepts a string `s` from the user and replaces all character `'3'` in `s` with character `'e'` then prints the converted string on the screen.

For example, if `s = "cod3l3arn"`, enter the following string:

```
cod3l3arn
```

When the code is compiled and executed, it produces the following result:

```
codelearn
```

Ex7. Given a string `s` and a character `c`. Write a program that accepts these two variables from the user and finds the first occurrence of `c` in string `s`. If character `c` does not appear in `s`, print `-1`.

For example, if `s = "codelearn"` and `c = 'o'`, enter the following values:

```
codelearn o
```

When the code is compiled and executed, it produces the following result:

```
1
```

Because the first occurrence of '`o`' is at position `1`

If you enter the following values:

```
codewar z
```

When the code is compiled and executed, it produces the following result:

```
-1
```

Because `'z'` does not appear in string `"codewar"`.

Ex8. Given a string `s`. Write a program that accepts a string `s` from the user and converts lowercase characters in `s` to uppercase characters then prints the converted string on the screen:

For example, if `s = "Codelearn"`, enter the following values:

```
Codelearn
```

When the code is compiled and executed, it produces the following result:

```
CODELEARN
```

Ex9. Write a program that accepts the names of two people from the user then checks whether these two names are the same or not.

If two names are the same, print the following line on the screen:

```
two people have the same name
```

If two names are different, print the following line on the screen:

```
two people don't have the same name
```

Ex10. Write a program to print all characters from `'A'` to `'Z'` as below:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Ex11. Consider the following programs

```
int main() {
    char s[10] = "?";

    scanf("%s", s);

    return 0;
}
```

```
int main() {
    char s[10] = "?";

    scanf("%[^\n]", s);

    return 0;
}
```

```
int main() {
    char s[10] = "?";

    gets(s);

    return 0;
}
```

```
void getstr(char [], int);

int main() {
    char s[10] = "?";

    getstr(s, 9);

    return 0;
}
```

```
void getstr(char s[], int max) {
    int i, c;

    i = 0;
    while((c = getchar()) != '\n' && c != EOF)
        if (i < max)
            s[i++] = (char) c;
    s[i] = '\0';
}
```

Fill in the following table to show the data stored in **s** by each of these programs

User Input	scanf("%s", s)	scanf("%[^\n]", s)	gets(s)	getstr(s, 10)
"Hi, there"				
"Hello, there"				
""				

Compare your answers with those of one of your collegues and make sure that the two of you agree.

Ex12. Design and code a function named **wordCount** that receives a null terminated string of text and returns the number of words contained in the string. Consider a word to be any sequence of non-whitespace characters.

The whitespace characters include newline, horizontal tab, form feed, vertical tab and space characters.

Finally, write a program that accepts a string to be counted and displays the number of words in the string. You may assume that the user will not input a string that contains more than 100 characters, but may input an empty string.

The output from your program looks something like like:

```
Word Counter
============
String to be counted :   BTP100   is not    that  hard!
Number of words in the string : 5
```