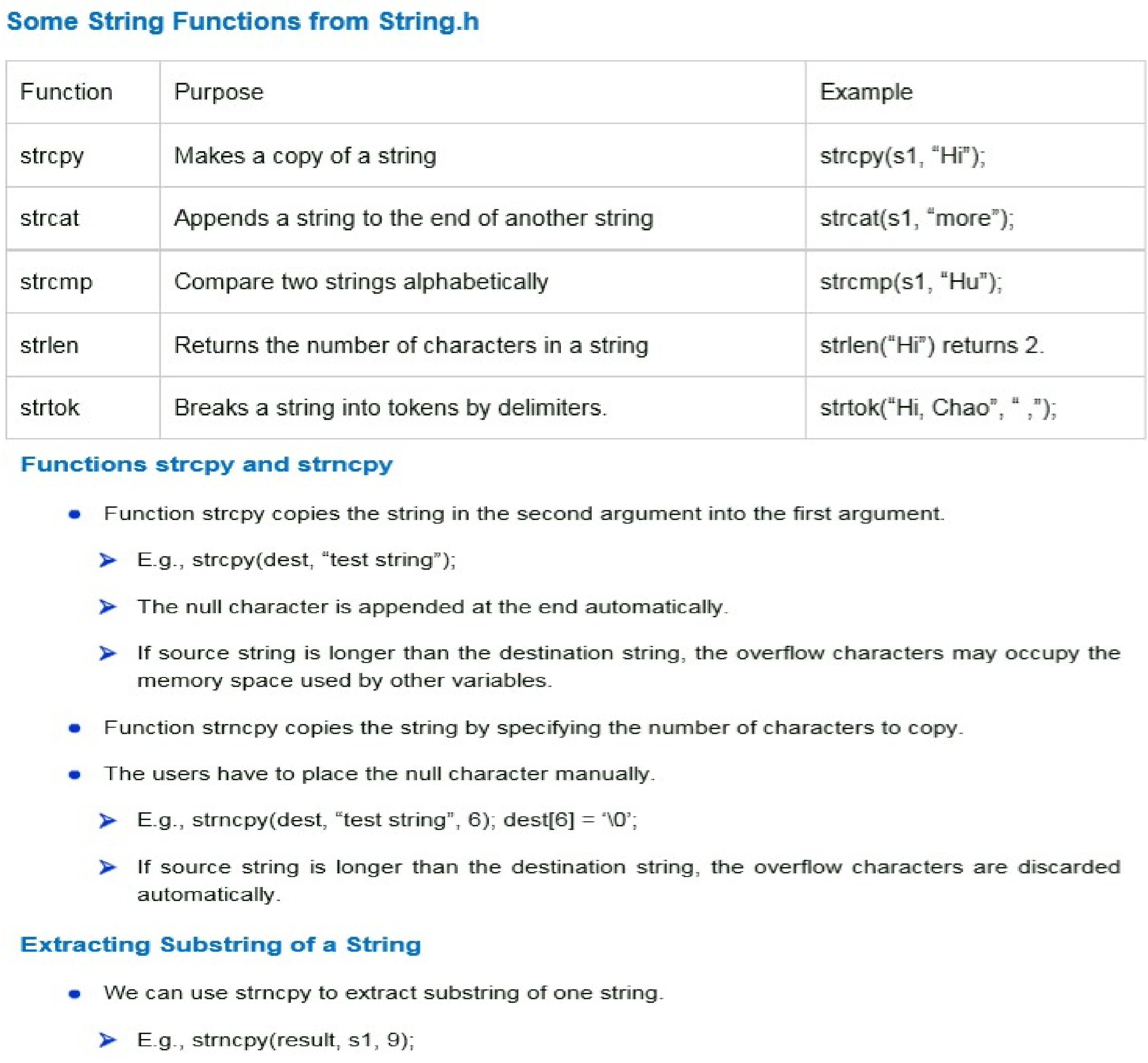
# Chapter - 5 : Array

### String Library Functions

- The string can not be copied by the assignment operator ‘=’.
- E.g, “str = “Test String”” is not valid.
- C provides string manipulating functions in the “**string.h**” library.



## Some String Functions from String.h

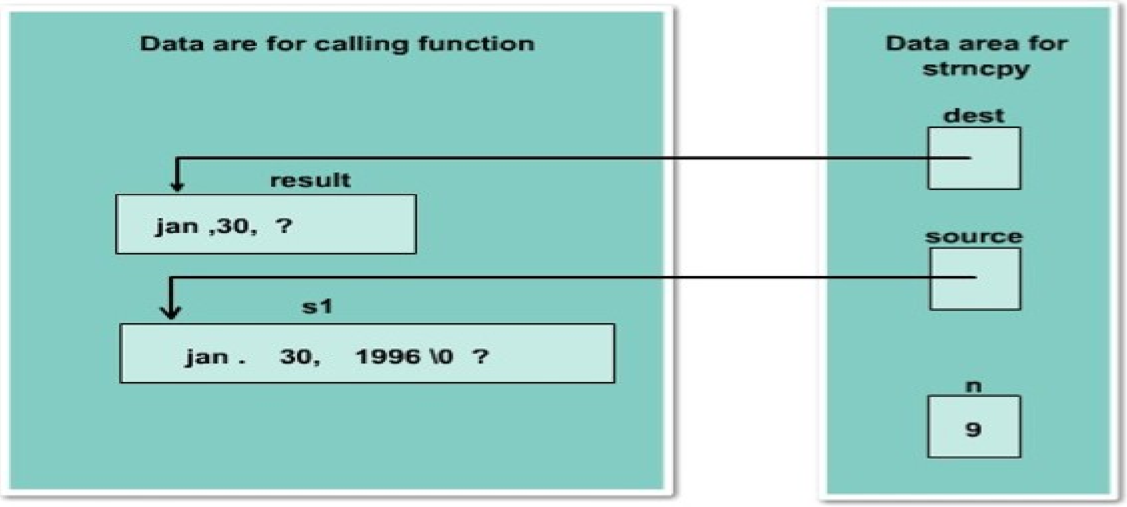
| Function | Purpose | Example |
|---|---|---|
| strcpy | Makes a copy of a string | strcpy(s1, “Hi”); |
| strcat | Appends a string to the end of another string | strcat(s1, “more”); |
| strcmp | Compare two strings alphabetically | strcmp(s1, “Hu”); |
| strlen | Returns the number of characters in a string | strlen(“Hi”) returns 2. |
| strtok | Breaks a string into tokens by delimiters. | strtok(“Hi, Chao”, “ ,”); |

### Functions strcpy and strncpy

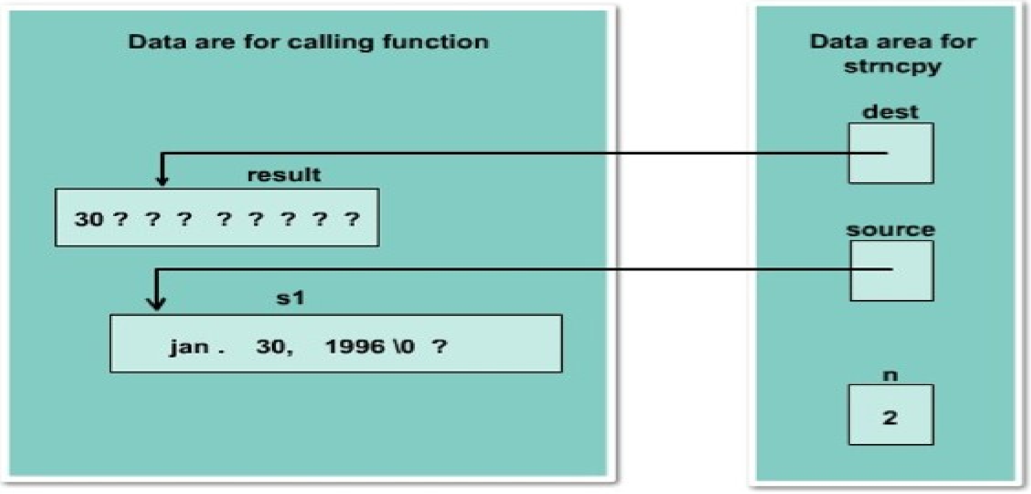
- Function strcpy copies the string in the second argument into the first argument.
  - E.g., strcpy(dest, “test string”);
  - The null character is appended at the end automatically.
  - If source string is longer than the destination string, the overflow characters may occupy the memory space used by other variables.
- Function strncpy copies the string by specifying the number of characters to copy.
- The users have to place the null character manually.
  - E.g., strncpy(dest, “test string”, 6); dest[6] = ‘\0’;
  - If source string is longer than the destination string, the overflow characters are discarded automatically.

### Extracting Substring of a String

- We can use strncpy to extract substring of one string.
  - E.g., strncpy(result, s1, 9);



- E.g., strncpy(result, &s1[5], 2);

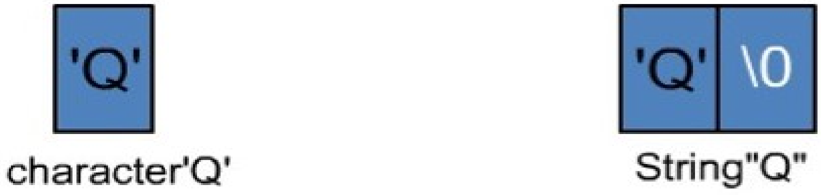


### Functions strcat and strlen

- Functions strcat and strncat concatenate the fist string argument with the second string argument.
  - strcat(dest, “more..”);
  - strncat(dest, “more..”, 3);
- Function strlen is often used to check the length of a string (i.e., the number of characters before the fist null character).
  - E.g., dest[6] = “Hello”;
  - strncat(dest, “more”, 5-strlen(dest));
  - dest[5] = ‘\0’;

## Distinction Between Characters and Strings

- The representation of a char (e.g., ‘Q’) and a string (e.g., “Q”) is essentially different.
  - A string is an array of characters ended with the null character.



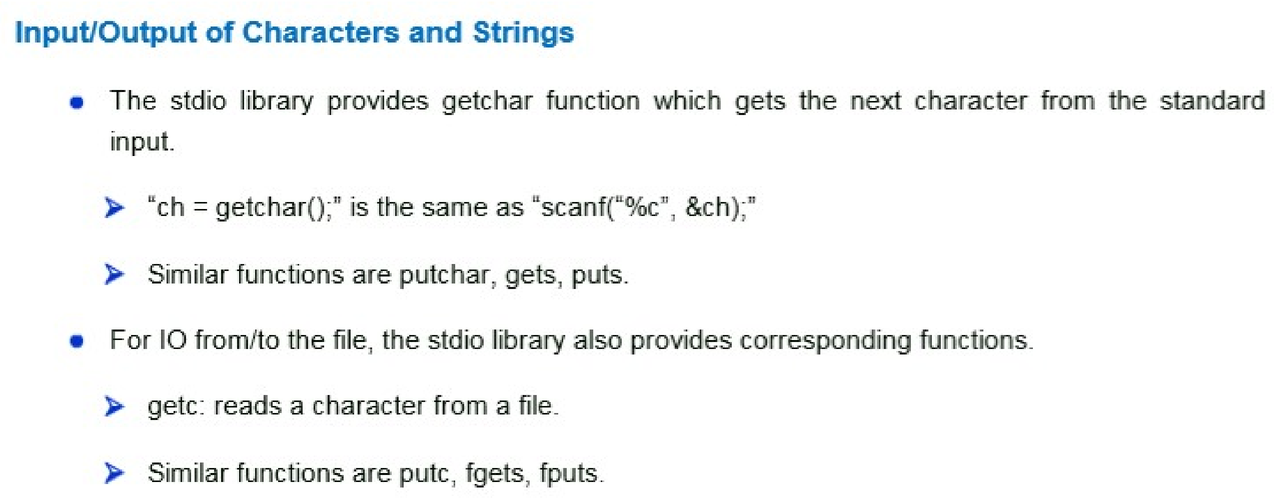
## String Comparison (1/2)

- Suppose there are two strings, str1 and str2.
  - The condition str1 < str2 compare the initial memory address of str1 and of str2.
- The comparison between two strings is done by comparing each corresponding character in them.
  - The characters are compared against the ASCII table.
  - “thrill” > “throw” since ‘i’ < ‘o’;
  - “joy” < joyous“;
- The standard string comparison uses the strcmp and strncmp functions.

## String Comparison (2/2)

| Relationship | Returned Value | Example |
|---|---|---|
| str1 < str2 | Negative | “Hello” < “Hi” |
| str1 = str2 | 0 | “Hi” = “Hi” |
| str1 > str2 | Positive | “Hi” > “Hello” |

- E.g., we can check if two strings are the same by
  - if(strcmp(str1, str2) != 0)
  - printf(“The two strings are different!”);



## Input/Output of Characters and Strings

- The stdio library provides getchar function which gets the next character from the standard input.
  - “ch = getchar();” is the same as “scanf(“%c”, &ch);”
  - Similar functions are putchar, gets, puts.
- For IO from/to the file, the stdio library also provides corresponding functions.
  - getc: reads a character from a file.
  - Similar functions are putc, fgets, fputs.

## Example of strlen:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[20] = "BeginnersBook";
    printf("Length of string str1: %d", strlen(str1));
    return 0;
}
```

**Output** : Length of string str1: 13

## Example of strcmp:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[20] = "BeginnersBook";
    char s2[20] = "BeginnersBook.COM";
    if (strcmp(s1, s2) ==0)
    {
      printf("string 1 and string 2 are equal");
    }else
     {
       printf("string 1 and 2 are different");
     }
    return 0;
}
```

## Output:

```
string 1 and 2 are different
```

**Example of strncmp:**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[20] = "BeginnersBook";
    char s2[20] = "BeginnersBook.COM";
    /* below it is comparing first 8 characters of s1 and s2*/
    if (strncmp(s1, s2, 8) ==0)
    {
       printf("string 1 and string 2 are equal");
    }else
    {
       printf("string 1 and 2 are different");
    }
    return 0;
}
```

**Output:**

```
string1 and string 2 are equal
```

**Example of strcpy:**

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[30] = "string 1";
    char s2[30] = "string 2 : I’m gonna copied into s1";
    /* this function has copied s2 into s1*/
    strcpy(s1,s2);
    printf("String s1 is: %s", s1);
    return 0;
}
```

**Output:**

```
String s1 is: string 2: I’m gonna copied into s1
```

**Example of strchr**

```
// C code to demonstrate the working of
// strrchr()

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

// Driver function
int main()
{

    // initializing variables
    char st[] = "GeeksforGeeks";
    char ch = 'e';
    char* val;

    // Use of strrchr()
    // returns "ks"
    val = strrchr(st, ch);

    printf("String after last %c is :  %s \n",  ch, val);

    char ch2 = 'm';

    // Use of strrchr()
    // returns null
    // test for null
    val = strrchr(st, ch2);

    printf("String after last %c is :  %s ",  ch2, val);

    return (0);
}
```

**Output:**

```
String after last e is :  eks

String after last m is :  (null)
```