Strings

Introduction

In C, a string is a null-terminated character array. This means the last character, a null character (’\0’) is stored to signify the end of the character array.

Reading Strings

Strings can be read by the user in three ways:

1. using `scanf` function,
2. using `gets()` function, and
3. using `getchar()` repeatedly.

Strings can be read using `scanf()` by writing

```
scanf("%s", str);
```

where `str` is an array of characters. The main pitfall of using `scanf()` function is that the function terminates as soon as it finds a blank space. You may specify a field width to indicate the maximum number of characters that can be read. Extra characters are left unconsumed in the input buffer.

The next method of reading a string is by using the `gets()` function. The string can be read by writing

```
gets(str);
```

The `gets()` takes the starting address of the string which will hold the input. The string inputted using `gets()` is automatically terminated with a `null` character.

Strings can also be read by calling `getchar()` function repeatdely to read a sequence of single characters (unless a terminating character is entered) and simultaneously storing it in a character array

```
i = 0;()
ch = getchar; // Get a character
while(ch != ’*’) {
     str[i] = ch; // Store the read character in str
     i++;
     ch = getchar(); Get another character
}
str[i] = ’\0’; // Get another character
```

Writing Strings

Stringgs can be displayed on the sceen using the following three ways:

1. using `printf()` function,
2. using `puts()` function, and
3. using `putchar()` function repeatedly.

Strings can be displayed using `printf()` by writing

```
printf("%s", str);
```

The next method of writing a string is by using the `put()` function. A string can be displayed by writing

```
puts(str);
```

Strings can also be written by calling the `putchar()` repeatedly to print a sequence of single characters.

```
i = 0;
while(str[i] != '\0') {
     putchar(str[i]); // Print the character on the screen
     i++;
}
```

Points To Remember

• A string is a null-terminated character array.

• Individual characters of strings can be accesed using a subscript that starts from zero.

• All the characters of a string are stored in successive memory locations.

• Strings can be read by a user using three ways: `scanf()` function, using `gets()` function, or using `getchar()` function repeatedly.

• The `gets()` function takes the starting address of the string which will hold the input. The string inputted using `gets()` is automatically terminated with a `null` character.

• Strings can also be read by calling `getchar()` repeatedly to read a sequence of single characters.

• Strings can be displayed on the screen using three ways: using `printf()` function, using `puts()` function, or using `putchar()` function repeatedly.

• C standard library supports a number of pre-defined functions for manipulating strings or changing the contents of strings. Many of these functions are defined in the header file `string,h`.

• Name of a string acts as a pointer to the string. In the declaration `str[5] = "hello";` `str` is a pointer which holds the address of the first character, i.e., 'h'.

• An array of strings can be declared as `char strings[20][30];` where the first subscript denotes the number of strings and the second subscript denotes the length of every individual string.