STRING

Advanced Tech. P – SVMC

Contents

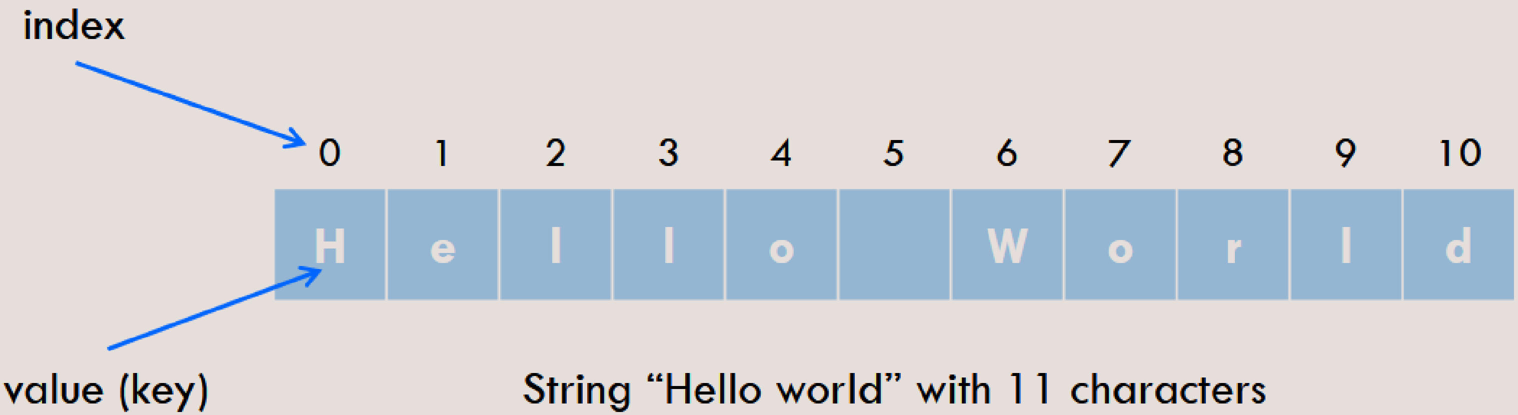
- String manipulation
- String library
- Palindrome & pattern matching
- Practice

String

- □ A character is a unit of information that roughly corresponds to a grapheme, grapheme-like unit, or symbol.

  Eg: letters (a-z, A-Z), numerical digits(0-9), common punctuation marks (".", "-"), and whitespace.

- ❑ A string is traditionally a sequence of characters.



0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		W	o	r	l	d

String "Hello world" with 11 characters

- ❑ **Note:** in C, a string is end at '\0'.

String Declaration

C/C++

```
char name[size];
char name[size] = "initial string";
```

Ex:

```
char S[50]; //a string with maximum 50 characters
char S[50] = "Hello World";
S[4] = 0; //S = ???
```

Java

```
char[] SS = {'i','n','i','t','i','a','l',' ','s','t','r','i','n','g'};
String S = new String(SS); //new keyword
String S = "initial string"; //string literal
```

Ex:

```
char[] SS = {'H','e','l','l','o',' ','W','o','r','l','d','!'};
String S = new String(SS);
String S = "Hello World!";
```

Contents

- **String manipulation**
- String library
- Palindrome & pattern matching
- Practice

String Length

- □ **C/C++**
- □ **Make function**

```
int my_strlen(char S[]){
    int len = 0;
    while (S[len] != 0) //S[len] != '\0'
        len++;
    return len;
}
```

- □ **Using STL** (strlen())

```
#include <stdio.h>
#include <string.h>
int main(){
    char S[] = "Hello World";
    printf("Length of string is: %d", strlen(S));
}
```

String Length

- □ ***Java***
- □ **Using STL** (length())

```
class String_Length{
    public static void main(String args[]) throws Exception{
        String S = "Hello World";
        System.out.println("Length of string is " + S.length());
    }
}
```

Accessing String Elements

- **C/C++**

```c
#include <stdio.h>
int main(){
    char S[] = "Hello World";
    int i = 0;
    while (S[i] != 0){
        if (S[i] != ' ')
            printf("%c", S[i]);
        i++;
    }
}
```

Accessing String Elements

- ***Java***

```
class String_Accessing{
    public static void main(String args[]) throws Exception{
        String S = "Hello World";
        for (int i = 0; i < S.length(); i++)
            if (S.charAt(i) != ' ')
                System.out.print(S.charAt(i));
    }
}
```

Exercise 1

□ Name normalization

Given a string of name, your goal is normalize this name as following:

1. The name must contain only letters and space characters.
2. The fisrt letter of each word must be in UPPER case, the other are in lower case.
3. No space in the begin and end of name.
4. Only one space between two words.

Eg: If you given a string “ *mY   Na99Me   i2s John1*   ”, you need to output “*My Name Is John*”

Contents

- String manipulation
- **String library**
- Palindrome & pattern matching
- Practice

String Copy

- □ **C/C++**
- □ **Make function**

```
void my_strcpy(char Dest[], char Source[]){
    int i = 0;
    while (Source[i] != 0){
        Dest[i] = Source[i];
        i++;
    }
    Dest[i] = 0;
}
```

```
void my_strcpy2(char* Dest, char* Source){
    while (*Source != 0)
        *Dest++ = *Source++;
    *Dest = 0;
}
```

String Copy

- □ **C/C++** (cont.)
- □ **Using STL** (strcpy())

```
#include <stdio.h>
#include <string.h>
int main(){
    char S[] = "Hello World";
    char SS[50];
    strcpy(SS, S);
    printf("%s\n", SS);
}
```

What's the output of the above program if we declare SS as follow?

```
    char SS[5];
```

String Copy

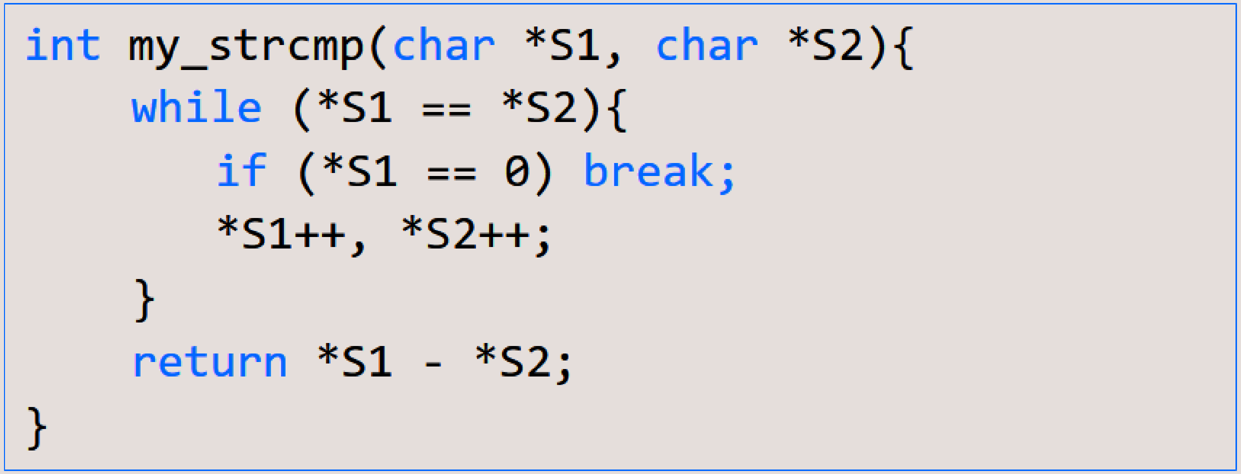
- ***Java***

```
String S = "Hello World";
String SS = S;
System.out.print(SS);
```

Very simple!!!

String Compare

- □ **C/C++**
- □ **Make function**



```
int my_strcmp(char *S1, char *S2){
    while (*S1 == *S2){
        if (*S1 == 0) break;
        *S1++, *S2++;
    }
    return *S1 - *S2;
}
```

What's the output of the above program?

- ❑ **Using STL** strcmp()
- ❑ ***Java***

Using methods equal(), compareTo(), compareToIgnoreCase().

String Reverse

- **C/C++**

```
void my_reverse(char *S){
    int len = strlen(S);
    char tmp;
    for (int i = 0; i < len/2; i++){
        tmp = S[i];
        S[i] = S[len-i-1];
        S[len-i-1] = tmp;
    }
}
```

- ***Java***

```
String S = "Hello World";
String SS = new StringBuffer(SS).reverse().toString();
System.out.print(SS);
```

String to number

- C/C++
- **Make function**

```
int my_atoi(char *S){
    int i = 0, val = 0;
    while (S[i] >= '0' && S[i] <= '9'){
        val = val*10 + S[i] - '0';
        i++;
    }
    return val;
}
```

What's output if we call `my_atoi("12345abc")` and `my_atoi("-123")`?

- **Using STL** atoi()
- ***Java***

Using methods Integer.*parseInt*(), Integer.*valueOf*().

Number to string

- **C/C++**
- **Make function**

```
void my_itoa(int val, char *S){
    int i = 0;
    while (val > 0){
        S[i++] = val % 10 + '0';
        val = val/10;
    }
    S[i] = 0;
    my_reverse(S);
}
```

What's output if we call `my_itoa(123, S)`, `my_itoa(-123, S)` and `my_itoa(-(-123), S)`

- **Using STL** itoa()
- ***Java***

Using method Integer.*toString()*.

Contents

- String manipulation
- String library
- **Palindrome & pattern matching**
- Practice

Exercise 2

- **Palindrome**

- A string is said to be palindrome if we write it from left and right then we get the same result. For example “non” is a palindrome of size 3.
- A substring of S is a string begin at character i^{th} and end at j^{th} of string S. For example, “el” is a subtring of “Hello”.

Give a string, your goal is find the longest palindrome substring of it.

Eg: “aabbaa” is the answer for case “aaabbaac”.

Exercise 3

- **Pattern Matching**

Given two strings S and P, you need to count how many subtring of S is P.

Eg: String "abaabbbabba" have 3 substring is "ab".

Contents

- String manipulation
- String library
- Palindrome & pattern matching
- **Practice**

Pratice

1. **To and Fro (400)** http://www.spoj.com/problems/TOANDFRO/
2. **Mirror Strings (12262)** http://www.spoj.com/problems/MSUBSTR/

Homework

1. **Anti-Blot System (2157)** http://www.spoj.com/problems/ABSYS/
2. **Broken Keyboard (2852)** http://www.spoj.com/problems/BROKEN/
3. **Find String Roots (7212)** http://www.spoj.com/problems/FINDSR/

Reference

- [wiki] Character (computing) https://en.wikipedia.org/wiki/Character_(computing)
- [wiki] String (computer science) https://en.wikipedia.org/wiki/String_(computer_science)