**String Built-in Functions:**

1. **Constructor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Details** | **Time C** |
| **isalpha(character)** | Returns a nonzero number if the character is a letter ('A' - 'Z', 'a' -  'z'); otherwise it returns zero. | O(1) |
| **isalnum(character)** | Returns a nonzero number if the character is a letter ('A' - 'Z', 'a' -  'z', or '0' - '9'; otherwise it returns zero. | O(N) |
| **isdigit(character)** | Returns a nonzero number if the character is digit (0 through 9);  otherwise it returns a zero. | O(N) |
| **isspace(character)** | Returns a nonzero number if the character is a whitespace (tab,  space, newline); otherwise it returns a zero. | O(N) |
| **isupper(character)** | Returns a nonzero number if the character is uppercase;  otherwise it returns a zero. | O(N) |
| **islower(character)** | Returns a nonzero number if the character is lowercase;otherwise it returns a zero. |  |
| **toupper(character)** | Return the uppercase equivalent if the character is lowercase;  otherwise it returns the character unchanged. |  |
| **tolower(character)** | Return the lowercase equivalent if the character is uppercase;  otherwise it returns the character unchanged |  |
| **stoi(string)** | Converts an ASCII string to an integer (include #<stdlib.h> in  your program) |  |
| **stof(string)** | Converts an ASCII string to an float (include #<stdlib.h> in your  program) |  |

1. **Capacity**

|  |  |  |
| --- | --- | --- |
| **Name** | **Details** | **Time Complexity** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Modifiers**

|  |  |  |
| --- | --- | --- |
| **Name** | **Details** | **Time Complexity** |
| **+= or append()** | string s *=* "Hello", t *=* "World";  s *+=* t; // s.append(t);  string s;  s.append(5, 'a'); // s = “aaaaa” |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Element access**

|  |  |  |
| --- | --- | --- |
| **Name** | **Details** | **Time Comp** |
| **find()** | string s = "LeetcodeCodeforceAtCoder", sub = "Leetcode";   auto chack = s.find(sub);  (chack != string::npos)? cout << "FOUND\n" : cout << "NOT FOUND\n"; |  |
| **erase()** | string s = "LeetcodeCodeforceAtCoder", sub = "Leetcode";  auto chack = s.find(sub);  if(chack != string::npos) s.erase(chack, sub.length());  string s = "hello word";  s.erase(6, 5); // output : s = hello |  |
| **substr()** | string s = "Hello\_Word";  string cpy = s.substr(6, 4);  cout << cpy << endl; // Output: Word  string s = "Hello\_Word";  auto start = s.find('e');  string cpy = s.substr(start+1); // Output: llo\_Word |  |
| **count()** | string s = "abcdaba";  int cnt = count(s.begin(), s.end(), 'a'); | How many ‘a” is here, it will be count; |
| **replace()** | string s = "hello word", m = "anmamun0";  s.replace(6, 5, m); // output : s = hello anmamun0 |  |
| **insert()** | string s = "hello ", m = "word ";  s.insert(2, m); // output: heword llo |  |
| **compare()** | string s = "hello", m = "world";  int result = s.compare(w);  if(result == 0) cout<<"Equal\n";  else if(result <0) cout<<"S comes before w (lexicographically) \n";  else cout<<"S comes after w\n"; |  |
| **clear()** | string s = "hello";  s.clear(); |  |
| **empty()** | string s = "hello";  (s.empty()) ? yes : no; |  |
| **size() or length()** | string s = "hello";  int sz = s.size(), l = s.length(); |  |
| **stringstream** | string s *=* "Hello World";  stringstream line(s);  string word;  *while*(line *>>* word) // (getline(line,word," "))  {} |  |
| **Remove first all similar char** | if(s[0]=='B')  {  int x = s.find\_first\_not\_of('B');     s.erase(0, x);  } | Intput: BBBAABB  Output: AABB |
| **Remove last all similar char** | ifand s.back()=='A')  {     int x = s.find\_last\_not\_of('A');     s.erase(x + 1);  }  //  size\_t x *=* s.find\_last\_not\_of('A');  *if* (x *!=* string::npos) s.erase(x *+* 1); | Input : BBABAAA  Output: BBAB |