

**Mawlana Bhashani Science and Technology University**

**Department of Information and Communication Technology**

**Assignment: 02**

**Assignment Name:** Functions of String

**Device info:**

System type: 64-bit operating system

Window Edition: Windows 11 Home Single Language

Code Blocks Version: Code::Blocks 20.03

|  |  |
| --- | --- |
| **Submitted By** | **Submitted To** |
| Name: Kuldip Saha Mugdha | Bikash Kumar Paul |
| ID: IT22018 | Assistant Professor |
| 1st Year 2nd Semester  Session: 2021-2022 | DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY  **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY** |
|  |  |

Date: 15-08-2023

**1.strlen():**

* The strlen() function in C is used to calculate the length of a string.

|  |  |
| --- | --- |
| **Code** | **Input & Output** |
| #include <stdio.h>  #include <string.h>    int main()  {        char Mugdha[100];   gets(Mugdha);          int l = strlen(Mugdha);      printf("Length of string is : %d\n", l);        return 0;  } |  |

**2.strnlen():**

* strnlen() returns the number of characters in the string s, not including the terminating \0 character, but at most maxlen.

|  |  |
| --- | --- |
| **Code** | **Input & Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char Kuldip[100];  gets(Kuldip);  printf("Length of string when maxlen is 10: %ld \n", strnlen(Kuldip, 10));  return 0;  } |  |

**3.strcmp():**

* It compares two strings and returns 0 if the strings are the same.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  int i;  char Kuldip[30],Saha[30];  gets(Kuldip);  gets(Saha);  i=strcmp(Kuldip,Saha);  if(i==0)  printf("Equal\n");  else  printf("Not equal\n");  return 0;  } |  |

**4.strncmp():**

* It compares two strings only to n characters.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  int main(){  char Kuldip[100],Saha[100];  scanf("%s",Kuldip);  scanf("%s",Saha);  if(strncmp(Kuldip,Mugdha,6)==0)  printf("Equal");  else  printf("Not equal");  }  return 0;  } |  |

**5.strcat():**

* It concatenates two strings and returns the concatenated string.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  int main(){  char Kuldip[10],Saha[10];  scanf("%s",Kuldip);  scanf("%s",Saha);  strcat(Kuldip,Saha);  printf("%s",Kuldip);  return 0;  } |  |

**6.strncat():**

* It concatenates n characters of one string to another string.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  int main(){  char Kuldip[10],Saha[10];  scanf("%s",Kuldip);  scanf("%s",Saha);  strcat(Kuldip,Saha,5);  printf("%s",Kuldip);  return 0;  } |  |

**7.strcpy():**

* It copies one string into another.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char Kuldip[30],Saha[30];  gets(Kuldip);  gets(Saha);  strcpy(Kuldip,Saha);  printf("%s\n",Kuldip);  return 0;  } |  |

**8.strncpy():**

* It copies the first n characters of one string into another.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include <stdio.h>  #include <string.h>  int main() {  char Kuldip[10],Saha[20];  scanf("%s%s",Kuldip,Saha);  strncpy(Kuldip,saha,10);  printf("String is:%s\n",Kuldip);  printf("String is:%s\n",Saha);  return 0;  } |  |

**9.strchr():**

|  |
| --- |
| * It finds out the first occurrence of a given character in a string. |
|  |

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include <stdio.h>  #include <string.h>  int main ()  {  char Kuldip[55] ="Kuldip Saha Mugdha";  char \*Saha;  Saha = strchr (Kuldip,'i');  printf ("Character i is found at position %d\n",Saha-Kuldip+1);  printf ("First occurrence of character \"i\" in \"%s\" is" \  " \"%s\"",Kuldip,Saha);  return 0;  } |  |

**10.strrchr():**

* It finds out the last occurrence of a given character in a string.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include <stdio.h>  #include <string.h>  int main () {  char Kuldip[] = "Kuldip-Saha-Mugdha";  char Saha = '-';  char \*Mugdha;  Mugdha = strrchr(Kuldip,Saha);  printf("String starting from last occurrence of %c is: %s\n",Kuldip,Mugdha);  return 0;  } |  |

**11.strstr():**

* It finds out the first occurrence of a string in a given string.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include <string.h>  #include <stdio.h>  int main()  {  char Kuldip[] = "Kuldip\_Saha";  char Saha[] = "Sah";  char\* Mugdha;  Mugdha = strstr(Kuldip,Saha);  if (Mugdha) {  printf("String found\n");  printf("First occurrence of string '%s' in '%s' is '%s'",Saha,Kuldip,Mugdha);  } else  printf("String not found\n");  return 0;  } |  |

**12.strcasecmp():**

* It compares two strings without sensitivity to the case.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include <stdio.h>  #include <string.h>  int main () {  char Kuldip[100],Saha[100];  scanf("%s %s",Kuldip,Saha);  int result=strcasecmp(Kuldip,Saha);  if (result==0)  printf("0\n");  else if (result < 0)  printf("-1\n");  else  printf("1\n");  } |  |

**13.strncasecmp():**

* It compares n characters of one string to another without sensitivity to the case.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main() {  char Kuldip[50],Saha[50];  int Mugdha;  gets(Kuldip);  gets(Saha);  Mugdha = strncasecmp(Kuldip,Saha,3);  if(Mugdha==0)  printf("Strings are equal.\n");  else if(Mugdha < 0)  printf("s1 is less then s2\n");  else  printf("s2 is less then s1\n");}} |  |

**14.strupr():**

* It converts a given string to uppercase.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char Mugdha[30];  gets(Mugdha);  strupr(Mugdha);  printf("%s\n",Mugdha);  return 0;  } |  |

**15.strlwr():**

* It converts a given string to lowercase.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char Mugdha[30];  gets(Mugdha);  strlwr(Mugdha);  printf("%s\n",Mugdha);  return 0;  } |  |

**16.strtok():**

* It is used to split string in multiple strings on the basis of delimiters.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char str[100]="Kuldip-Saha-Mugdha";  char\* token=strtok(str,"-");  while(token!=NULL) {  printf("%s\n", token);  token=strtok(NULL,"-");  }  return 0;  } |  |

**17.strrev():**

* It is used to reverse a string.

|  |  |
| --- | --- |
| **Code** | **Input/ Output** |
| #include<stdio.h>  #include<string.h>  int main(){  char Mugdha[14];  gets(Mugdha);  strrev(Mugdha);  printf("%s",Mugdha);  return 0;  } |  |

**18.Sscanf()**

* The C library function **int sscanf(const char \*str, const char \*format, ...)** reads formatted input from a string.

|  |  |
| --- | --- |
| **Code** | **Input/Output** |
| #include <stdio.h>  int main() {  char Kuldip[50];  scanf(" %[^\n]", Kuldip);  char name[50];  int age;  sscanf(Kuldip, "%s %d", name, &age);  printf("Name: %s\nAge: %d\n", name, age);  return 0;  } |  |

1. **Sprintf()**

* sprintf stands for **“String print”**. Instead of printing on console, it store output on char buffer which are specified in sprintf.

|  |  |
| --- | --- |
| **Code** | **Input/Output** |
| #include <stdio.h>  int main()  {  char Kuldip[100];  char Saha[50]= "Kuldip Saha Mugdha";  sprintf(Kuldip,"My name is: %s", Saha);  printf("%s\n", Kuldip);  return 0;  } |  |

**20.strcsp()**

* Returns the span of the source string not containing any character of the given string.

|  |  |
| --- | --- |
| **Code** | **Input/Output** |
| #include <stdio.h>  #include <string.h>  int main()  {  int size;  char str1[] = "KuldipSaha";  char str2[] = "Mugdha";  size = strcspn(str1, str2);  printf("The unmatched characters before first matched character : %d\n", size);  } |  |

**21.strspn()**

* Returns the span of the source string containing only the characters of the given string.

|  |  |
| --- | --- |
| **Code** | **Input/Output** |
| #include <stdio.h>  #include <string.h>  int main () {  int len = strspn("Kuldip Saha","Kuldip");  printf("Length of initial segment matching : %d\n", len );  return(0);  } |  |