**7.STRINGS**

**Strings and their representation**

* String is defined as a sequence of characters.
* In terms of c language string is an array of characters.
* While storing the string in character array the size of the array must be one more then the size of the string. Because the last character in the string is ‘\0’ or NULL. The string terminated with NULL or ‘\0’ is known as null terminated string.
* For example: To store “HELLO” in array the array must be declared as char a[5].
* **For example:**
  + - String “HELLO” is stored as shown in fig

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ‘H’ | ‘E’ | ‘L’’’ | ‘L’ | ‘O’ | ‘\0’ |

* **String Character set:**
  + Lower case: a to z
  + Upper case: A to Z
  + Number: 0 to 9
  + Special Characters: + - \* % / ( ) [ ] { } $ # &,. ? ““‘‘@ Etc.
* Now we can assign a value to the string as

**char name[10]=”ketan”**

**OR char name[10]={‘k’,’e’,’t’,’a’,’n’,’\0’}; This is static assignment of string.**

The memory representation of this string is as

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ‘k’ | ‘e’ | ‘t’ | ‘a’ | ‘n’ | ‘\0’ | - | - | - | - |

Name:

**For Dynamic allocation we use,**

**gets(string name); OR scanf(“%s”, string name);**

**eg. gets(name); OR scanf(“%s”,name);**

**Various string manipulation functions**

* **String Manipulation Functions:**
  1. String length: this function find the length of the string
  2. String copy: this function copy one string in to another string
  3. String concate: this function concate two string and store it in to the another string
  4. String compare: this function compare two string
  5. substring: this function finds one string into the another string
  6. Reverse string: this operation is used to reverse the given string.
* **Algorithm to find length of string.**

**SLEN(str)**

* This function finds the length of the string.
* **Str** is the string whose length is to be find
  1. [initialize]

Count🡨0

* 1. [Process until end of the string]

Repeat step 3 until str[count] !=NULL

* 1. [increment counter]

Count🡨count + 1

* 1. [Finish]

Return count

* **Algorithm to copy one string in to another string**

**SCOPY(source, destination)**

* This function copies the string source in to the string destination.
  1. [initialize]

Count🡨0

* 1. [Process until end of the loop]

Repeat step 3 until source [count] !=NULL

* 1. [copy character by character and increment counter]

Destination [count] 🡨source [count]

Count🡨count+1

* 1. [finish]

Destination [count]🡨NULL

* **Algorithm to concate two strings.**

**SCAT(s1,s2,s3)**

* This function concate two strings s1 and s2 in to the string s3
  1. [initialize]

count1🡨0

count2🡨0

* 1. [process until end of first string]

Repeat step 3 until s1 [count1] !=NULL

* 1. [copy character by character and increment counter]

s3 [count1] 🡨s1 [count1]

count1🡨count1+1

* 1. [process until end of second string]

repeat step 5 until s2 [count2] !=NULL

* 1. [copy character by character and increment counter]

s3 [count1]🡨s2[count2]

count1🡨count1+1

count2🡨count2+1

* 1. [finish]

s3 [count1]🡨 NULL

**for compare two string**

* **SCOMP(s1,s2)**
* This function compare two strings s1 and s2 and find weather they are equal or not.
  1. [initialize]

I🡨0

FLAG🡨0

* 1. [Find Length of two strings]

L1🡨strlen (s1)

L2🡨strlen (s2)

* 1. [Compare Length of two string]

If (L1 != L2) then

Write “Strings are not equal”

* 1. [Process string]

Repeat step 5 while I < L1

* 1. [compare character by character]

If s1 [I] !=s2 [I] then

FLAG🡨1

Else

I🡨I + 1

* 1. [Are string equal?]

If FLAG = 0 then

Write “Strings are equal”

Else

Write “Strings are not equal”

* 1. [Finished]

**Algorithm to find whether a second string is a substring of first string**

* **SSTR(S1, SUBSTR, ST,NUM)**
* This function finds the string **SUBSTR** starting at position ST in given string followed by NUM characters.
* Here **ST** indicates the starting position in given string.
* **NUM** indicates number of character after starting position.
  1. [initialize]

I🡨0

ST🡨ST-1

* 1. [Process the string]

Repeat step 3 until NUM < 0

* 1. [Compare Length of two string]

SUBSTR [I] 🡨S1 [ST]

I🡨I+1

ST🡨ST+1

NUM🡨NUM-1

* 1. [Finished]

SUBSTR [I] 🡨NULL

**Algorithm to reverse the string.**

* **REVERSE(org, rev)**
* This function reverses the given string and stores it in rev string.
  1. [Find Length of given string]

L🡨strlen (org)

* 1. [Initialize]

COUNT1🡨L-1

sCOUNT2🡨0

Rev [L] 🡨NULL

* 1. [Process until end of the string]

Repeat step 4 until org [COUNT2] ≠ NULL

* 1. [Copy Character by character and increment counter]

Rev [count1] 🡨org [COUNT2]

COUNT1🡨COUNT1-1

COUNT2🡨COUNT2+1

* 1. [Finished]

Write (rev)

**Algorithm for duplicating a given character string N times.**

**For example: Given the string “EXAM” and N=2 algorithms should generate the string “EXAMEXAM”.**

* **DUPLICATE(S1, S2, N)**
* This function duplicates the string S1 N times and stores the result string in S2.

1 [Initialize]

COUNT1🡨0

COUNT2🡨0

2 [Loop N times]

Repeat thru step 5 for I = 1 to N

3 [Process string]

Repeat step 4 until S1 [COUNT1] ≠ NULL

4 [Copy Character by character]

S2 [COUNT2]🡨S1[COUNT1]

COUNT1🡨COUNT1+1

COUNT2🡨COUNT2+1

5 [Reinitialize counter for S1]

COUNT1🡨0

6 [Finished]

S2 [COUNT2]🡨 NULL

**Algorithm for finding a palindrome string using recursion.(i.e RADAR)**

* **PALINDROME(FIRST,LAST,STRING)**
* FIRST indicate first element in the string.
* LAST indicate the last element in the string.
  + 1. [Find length of the string]

L1🡨strlen (STRING)

* + 1. [initialize]

FIRST🡨0

LAST🡨L1-1

* + 1. [compare elements]

If STRING [FIRST] ≠ STRING [LAST] then

Return 0;

Else if STRING [FIRST] = STRING [LAST] then

Call PALINDROME (FIRST, LAST, STRING)

Return 1

Else if STRING [FIRST] > STRING [LAST] then

Return 1

* + 1. [Check for return value]

If PALINDROME (FIRST, LAST, STRING) = 1 then

Write “String is palindrome”

Else

Write “String is not palindrome”

* + 1. [Finished]

**String conversion**

* The ‘c’ support the function that convert a string of digits in to their integer values.

**x a to i(string) ( Where x is an integer variable and string is a character array.)**

* Example: **number = ”1972”**

**Year= a to i (number)**

**Here a to i function convert string “1972” into integer 1972 and store into integer year.**

* There are a few functions that exist to convert strings to integer, long integer and float values.

**double atof(char \*string) -- Convert string to floating point value.   
int atoi(char \*string) -- Convert string to an integer value   
int atol(char \*string) -- Convert string to a long integer value.**

* For example:

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**char str[20]="55.444";**

**int i;**

**float f;**

**i=atoi(str);**

**f=atof(str);**

**printf("string to integer=%d",i);**

**printf("\nstring to float=%f",f);**

**getch();**

**}**

Output: **string to integer=55**

**String to float=55.444000**

**String Arrays**

* The array of string are defined as ,

**char name[3][10];**

* + These can be store three string of maximum length 10.
  + **For Example:**

**#include<stdio.h>**

**#include<conio.h>**

**void main()**

**{**

**char n[3][10]={"rutvik","rani","priya"};**

**clrscr();**

**printf("\n1st string=%s",n[0]);**

**printf("\n2nd string=%s",n[1]);**

**printf("\n3rd string=%s",n[2]);**

**getch();**

**}**

**Output:** 1st string= rutvik

2nd string=rani

3rd string=priya

* If we define a array of string like,

**char \*name[3];**

Which indicate that, store three string of any length.

**String programs**

* **Write a program to find the length of string.**

#include<stdio.h>

#include<conio.h>

void main()

{

char string1[10];

int count=0;

clrscr();

printf("enter the string:");

scanf("%s",&string1);

while(string1[count]!=NULL)

{

count=count+1;

}

printf("The length: %d ", count);

getch();

}

**Output:**

**Enter the string:** ANKIT

**The length:**5

* **Write a program to copy one string to another string.**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[10],str2[10];

int count=0;

clrscr();

printf(“enter the string”);

scanf(“%s”,str1);

while(str1[count]!=NULL)

{

str2[count]=str1[count];

count=count+1;

}

printf(“the copyied string=%s”,str2);

getch();

}

**Output:**

**Enter the string** MODI

**The copyied string=**MODI

* **Write a program to concate two string into another string.**

#include<stdio.h>

#include<conio.h>

void main()

{

char str1[10],str2[10],str3[20];

int count1=0, count2=0;

printf("\n Enter First String: ");

scanf("%s",&str1);

printf("\n Enter Second String: ");

scanf("%s",&str2);

while(str1[count1]!=NULL)

{

str3[count1]=str1[count1];

count1+=1;

}

while(str2[count2]!=NULL)

{

str3[count1]=str2[count2];

count1+=1;

count2+=1;

}

str3[count1]='\0';

printf("The concated String is : %s",str3);

getch();

}

* **Write a program to compare two strings.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char str1[5],str2[5];

int l1,l2,i=0,flag=0;

printf("enter string1");

scanf("%s",&str1);

printf("enter the string2");

scanf("%s",&str2);

l1=strlen(str1);

l2=strlen(str2);

if(l1!=l2)

printf("both are not equal");

else

{

for(i=0;i<l1;i++)

{

if(str1[i]!=str2[i])

flag=1;

}

if(flag==1)

printf("both are not equal");

else

printf("string are equal");

}

getch();

}

* **Write a program to subtract string to given length to given position.**

#include<stdio.h>

#include<conio.h>

void main()

{

int i=0,st,number;

char s1[10],substr[10];

clrscr();

printf("enter the string");

scanf("%s",s1);

printf("enter the start position from which u want new string");

scanf("%d",&st);

printf("enter the length of new string");

scanf("%d",&number);

st=st-1;

for(i=0;i<number;i++)

{

substr[i]=s1[st];

st=st+1;

}

substr[i]=NULL;

printf("the substrting string=%s",substr);

getch();

**}**

* **Write a program to reverse the string.**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char stro[5],strr[5];

int count1,count2=0,l;

printf("enter the original string:");

scanf("%s",stro);

l=strlen(stro);

printf("length=%d",l);

count1=l-1;

while(count2<l)

{

strr[count2]=stro[count1];

count2=count2+1;

count1=count1-1;

}

strr[count2]=NULL;

printf("the reverse string=%s",strr);

getch();

}

* **Write a program to duplicate the string to given time.**

#include<stdio.h>

#include<conio.h>

void main()

{

char s1[5],s2[20];

int n,i,c1=0,c2=0;

printf("enter the string:");

gets(s1);

printf("how many times u want to copy:");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

while(s1[c1]!=NULL)

{

s2[c2]=s1[c1];

c1=c1+1;

c2=c2+1;

}

c1=0;

}

s2[c2]=NULL;

printf("copy string=%s",s2);

getch();

}

* **Write a program to check string is palindrome or not without recursive.**

#include<stdio.h>

#include<conio.h>

#include<process.h>

void main()

{

char s1[10];

int la,f=0,i;

clrscr();

printf("enter the string:");

scanf("%s",s1);

la=strlen(s1)-1;

for(i=f;i<la;i++)

{

if(s1[i]==s1[la])

{

la--;

}

else

{

printf("string are not palindrom:");

getch();

exit(0);

}

}

printf("\nThe String is Palindrom");

getch();

}

* **Write a program of palindrome using recursive method.**