**STRINGS**

* Collection of characters is known as string.
* FREE() CODE :

#include<stdio.h>

#include<stdlib.h>

int main()

{

int \*ptr=NULL;

ptr = (int \*)malloc(3\*sizeof(int));

free(ptr);

// free(ptr);

printf("\n\n");

return 0;

}

**ERROR :**

**free() : Double free detected in tcache 2**

**Aborted (Core dumped)**

**>>**When we get double free. To avoid we have to see the code and remove double free which is same.

* Syntax for character array :

char Name[5] = {'H','e','l','l','o'};

* The string will start at the initial index and end at the null point because in ascii values the first base element is null. So, null is used at the ending of the string.
* String Syntax :

char Name[6] = {'H','e','l','l','o','\0'};

char Name[6] = "Hello";

char Greet[] = "Hello World";

Static array >> char Str1[21];

* In a two dimensional array, column is necessary but not the row.
* Code :

#include <stdio.h>

int main()

{

char Name[5];

int i;

for(i=0;i<20;i++)

scanf("%c",&Name[i]);

Name[5-1] ='\0';

for(i=0;i<20;i++)

printf("\n%c=%d",Name[i],Name[i]);

printf("\n\n");

printf("\n%s\n",Name);

puts(Name);

printf("\n\n");

return 0;

}

**STRCPY :**

* The first copy should be string.
* It will copy only the characters but not the null.

**STRCMP :**

* String compare.
* str1<Str2 == negative
* Str1>Str2 == positive
* Str1==str2 == 0
* **memcpy** is similar to strcpy but it deals with the memory
* **memset** to initialize
* **strstr**
* **strtok\*\*\*\*\***

**STRLEN :**

* string length

**STRERROR :**

* String error

**STRSEP :**

**STRSTR() :**

* To find the first occurrence of the substring needle in the string haystack.

**STRCHR**

Code :

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main()

{

char s1[20],s2[20];

char d1[40],d2[40];

char \*ptr = NULL;

int ret=0;

scanf("%s%s",s1,s2);

printf("\ns1=%s\ts2=%s",s1,s2);

//ptr = (char \*)malloc(strlen(s1)+1);

/\*

ret = strcmp(s1,s2);

printf("\nRet = %d",ret);

\*/

ptr = strcpy(d1,s1);

// printf("\nd1=%s",d1);

printf("\nptr = %s",ptr);

printf("\nAddress of d1=%u\nAddress of ptr=%u\n",&d1[0], ptr);

printf("\n\n");

return 0;

}

* **STRTOK CODE :**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char s1[400], s2[20];

char \*ptr = NULL;

printf("\nEnter the line with delimeter\n");

scanf("%[^\n]s",s1);

ptr = strtok(s1,",");

printf("\nptr = %s",ptr);

printf("\n\n");

return 0;

}

p6.c

ASSIGNMENT :

* To reverse a given string
* To reverse a given intervals of the string

i/p : k=3

Str1 = "abcdefg"

o/p

cbafedg

CODE :

struct, union, enum

struct tagName

{

members of structure

};

struct Square

{

int len;

int breadth;

}s4,s5,s6,s[10];

typedef struct Square SQR;

SQR s7,s8,s9;

typedef struct Square

{

int len;

int breadth;

}

SQR1;

SQR s1;

struct Chair

{

int noLegs;

char make[20];

char material[20];

char clr[20];

float price;

char DOM[20];

char placeManu[20];

};

**Struct \*\***

**s1.len**

1. **. => static var => nameVar.memberName**
2. **-> => ptr var => nameVar -> memberName**

SQR1 \*ptr;

ptr->len;

* n structure a function cannot be defined. Only fundamental datatypes or user defined datatypes can be defined.
* When a structure is defined it is advised to define it in a header file for easy and global access.
* CODE :

/\*  
demo on structures

\*/

#include<stdio.h>

struct Emp

{

int eId;

char eName[20];

float eSal;

char eGender;

char eAddress[20];

}