**DAY 11(8-11-2024)**

* Malloc have only one input but in calloc have two inputs
* Calloc is mostly widely used than malloc
* Free()
* If we already did free() then again when you do it shows error i.e

Free():double free detected in tcache 2 aborted (core dumped) ,so don’t do double free

* **Strings:**
* String is collection of characters.
* String is ends with NULL value (\0)
* Null value we do not use in any program so to differentiate we use the null value at the end of value
* With character no null is required
* Declaration with initialization

Char Name[6]={‘h’,’e’,’l’,’l’,’0’,’/0’}

=>give one size extra for null here

Char Name[6]=”hello”;

Char Name[]=”hello world”;

* Declaration without initialization:

Char Str[21];

* In 2D array columns are necessary but not rows.

Char Name[][20]={“ramya”,”Sowmya”,”mani”,”prasad”,”sai”};

Here there are 5 rows, in each word we have to use less than 20 characters (19 characters +\0);

Char Names[10][50]; 🡺 there are 10 names each of max capacity of 50 characters

* In string ,scanf (“%d”,Name); => here we donot mention “&Name” because in pointers thumb rule it shows &\* is nullify then

&\*(Name+0)🡺Name

* #include<stdio.h>

int main()  
{  
    char Name[5];  
    int i;  
    for (i=0;i<5;i++)  
        scanf("%c",&Name[i]);  
    for(i=0;i<5;i++)  
        printf("\n%c=%d",Name[i],Name[i]);  
  
    scanf("%s",&\*(Name+0));// scanf("%s",Name);  
    printf("%s",Name);//putc(name);  
    printf("\n\n");  
    return 0;  
  
}

* In this example ,In the size is 5 but the loop is till the 20 so it is illegal so it throws an error of stack smashing

**String Functions:**

* **Strcpy:**
* Strcopy is not copy the null values
* Strcpy is have the destination and source
* The copy one should not contain null value
* Strncpy is present
* Syntax:

Char \*strcpy(char \*dest,const char \*src);

* **Strcat:**
* Destination size have to be more than the source size
* Used to concentrate the two strings
* Strcat(s1,s2) 🡺 s1 is appended with s2
* S1=”ramya”,s2=”balivada”

Strcat(s1,s2)

New value of s1=ramyabalivada(without spaces)

S2=balivada(same as before)

* **Strcmp:**
* It compare the string by character by character
* It stop at when the character in the string is uncomparable
* Eg: ramya ranya
* It only checks until the m=!n so it does not check remaining character
* It returns the integer if s1>s2 it gives positive

S1<s2 => negative

S1=s2 => equal

Here it compares the ASIIC value of the characters

Eg:

* If(strcmp(s1,s2)==0)
* If((strcmp(e1.fname,”ramya”)==0) &&(strcmp(e1.lname,”Sowmya”)==0)
* If((expr1)&&(expr2))
* If((e1.id==101)&&(strcmp(s1,s2)==0))
* **In sprint memset and memcpy,strchr,strtok**
* **Strlen:**
* It length of the string excluding null character
* Strerror -captures string handling errors
* Int strlen(obj)

Printf(“%d”,strlen(name));

It gives the length except null character

* **Strstr:**
* Find the first occurrence of the substring needle in string haystack
* **Strchr:**
* Find the first occurrence of the character in string
* Strrchrr : Find the last occurrence of the character in string

* **Strtok:**
* It checks until the given delimiter and then it will print that string till that delimiter.
* It have own initial poniter

#include <stdio.h>

#include<string.h>

int main()

{

char s1[400],s2[20];

char \*ptr=NULL;

printf("\n enter the line with delimiter");

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

ptr=strtok(s1,",");

printf("ptr:%s\n",ptr);

printf("\n");

}

**Program :**

A screenshot of a computer code

Description automatically generated

Program3:

A screenshot of a computer program

Description automatically generated

A white screen with black text

Description automatically generated

Q1) to reverse a string

Q2) to reverse a given intervals in a string

* USER DEFINED DATATYPES:

1)structures

2)unions

3)enums

1)structures:

* These user defined data type which have the different data types
* If we want to create multiple data records
* Syntax:

Struct tagName

{

Members of structure

};

* Eg:

Struct Square

{

int len;

Int breadth;

};

* Declaration ;

Int a1,a2;

Struct square s1,s2,s3,s[10]; or

Struct Square

{

int len;

Int breadth;

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

* typedef struct Square SQR;

🡺short name for square we defined as SQR(in must be in capital letter)

Declaration for this is 🡺 SQR s7,s8,s9;

Or

typedef struct Square

{

int len;

Int breadth;

} SQR1;

mostly we use typedef is used

* Two ways to access the structure:

1. . used for static variable (nameVar.memberName)

Eg :

SQR s1

s1.breadth

1. -> used for pointer variable(nameVar->memberName)

Eg :

SQR1 \*ptr

Ptr->breadth

* In structure we should not define a function. only user defined can be defined.
* Structure usually declare globally outside the main and also can be declared in main also
* all the numeric are declared in the starting or character at the end or vice versa for avoid structure padding
* Structure also apply 8 bytes

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated