**STRINGS**

Char a=’D’

Char Name[5]={‘H’,’E’,’L’,’L’,’O’};-----collection of characters.

**String:** Collection of characters ended at null.

**Initialisation:**

Char Name[6]= {‘H’,’E’,’L’,’L’,’O’,’\0’};

Char Name[6]=”Hello”

Char Name[]=”Hello World”;-------size=no.of characaters+null

Char str1[21];

Char Name[][20]={“A”,”B”,”C”,”D”}-----names with 20 characters

Char Name[10][50]---there are 10 names each of max cap of 50 characters.

**EX:**

#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);

printf("%s",Name);

printf("\n");

return 0;

}

**ERROR:** stack smashing detected----scaning the characters beyond the size.

#include <stdio.h>

int main() {

char Name[5];

int i;

for(i=0;i<5;i++)

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

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

for(i=0;i<5;i++)

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

printf("\n\n");

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

puts(Name);

printf("\n");

return 0;

}

**Strncpy, strcpy -------**copy the certain characters

**strcmp,strcmp------**comparing the strings

**memset-----**to intialise the structures

**memcpy**---

**strtok**----tokenizing of the strings

**strlen----**length of the string except the null

**strerror**----to handle the string handling errors

**str str----**to find the first occurrence of string from main string

**strchar----**to find the first occurrence of character from main string

**strrchar-----**to find the last occurrence of character from main string

**STRING COMPARISION:**

#include <stdio.h>

#include<string.h>

int main() {

char s1[20],s2[20];

char d1[40],d2[40];

int ret=0;

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

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

ret=strcmp(s1,s2);

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

printf("\n\n");

return 0;

}

**EX:**

#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);

strcpy(d1,s1);

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

ptr=strcpy(d1,s1);

printf("\ns1=%d",d1);

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

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

printf("\n\n");

return 0;

}

**EX:**

#include <stdio.h>

#include<string.h>

int main() {

char s1[20],s2[20];

char d1[40],d2[40];

char \*ptr=NULL;

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

strcpy(d1,s1);

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

strcat(d1,s2);

printf("\nd2=%s",d1);

printf("\n\n");

return 0;

}

**STRTOK:**

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

int main() {

char s1[40],s2[40];

char \*ptr=NULL;

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

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

ptr=strtok(s1,",");

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

printf("\n\n");

return 0;

}

**EX:**

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

int main() {

char s1[40],s2[40];

char \*ptr=NULL;

char Names[10][20];

int count=0,i;

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

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

/\* ptr=strtok(s1,",");

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

ptr=strtok(NULL,",");

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

ptr=strtok(NULL,",");

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

ptr=strtok(NULL,",");\*/

ptr=strtok(s1,",");

count=0;

while(ptr!=NULL)

{

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

strcpy(Names[count],ptr);

count++;

ptr=strtok(NULL,",");

}

printf("\nNames are\n");

for(int i=0;i<count;i++)

printf("\n%s",Names[i]);

/\*if(ptr!=NULL)

printf("\nEnd of program");

else

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

printf("\n\n");

return 0;

}

**Write a program to reverse a string?**

**EX:**

#include <stdio.h>

#include<string.h>

int main() {

char arr[]="abcdefj";

int len=strlen(arr);

printf("%dLength of the string:\n",len);

printf("\nString: ");

for(int i=0;i<len;i++)

{

printf("%c",arr[i]);

}

printf("\n");

printf("\nReverse String: ");

for(int i=len-1;i>=0;i--)

{

printf("%c",arr[i]);

}

return 0;

}

**Write a program to reverse a given intervals of time?**

**EX:**

#include <stdio.h>

#include<string.h>

#include<stdlib.h>

void changeChars(char\* arr,int start,int end)

{

while(start<end)

{

int temp=arr[start];

arr[start]=arr[end];

arr[end]=temp;

start++;

end--;

}

}

void changeChar(char\* arr,int size,int k)

{

for(int i=0;i<size;i+=k)

{

int end=i+k-1;

changeChars(arr,i,end);

}

}

void printArr(char\* arr,int size)

{

for(int i=0;i<size;i++)

{

printf("%c ",arr[i]);

}

printf("\n");

}

int main() {

char arr[]="abcdefghi";

int size=strlen(arr);

int k=3;

printArr(arr,size);

changeChar(arr,size,k);

printArr(arr,size);

return 0;

}

**UDT(User Defined Datatypes):**

**Structures**

**Unions**

**Enums**

**Defined as----struct, union, enum**

**Syntax:**

Struct tagName

{

Members of structure

};

struct Square

{ int len;

Int breadth;

};

struct Chair

{ int noLegs;

char make[20];

char material[20];

char clr[20];

float price;

char DOM[20];

};

int a1,a2;

struct Square s1,s2,s3,sq[10];

struct Square

{ int len;

Int breadth;

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

**typedef struct Square SQR;-----we have to use only capital for typedef**

**SQR s7,s8,s9;**

typedef struct Square

{

int len;

int square; }SQR1;

**SQR1 s1;**

**S1.len----------------for normal variable**

1). ----if it is a static variable🡪 nameVar.memberName

2)->(if it is a pointer variable)🡪nameVar-> memberName---------these are the two ways to access

**SQR \*ptr;----ptr is a special variable pointing to SQR.**

**ptr->len;-------for pointer variable**

**Structures does not have a functions.**

**EX:**

#include <stdio.h>

#include <string.h>

struct Employee

{

char eName[20];

char eGender;

char eAddress[20];

int eId;

float eSal;

};

typedef struct Employee EMP;

void getDetails(EMP [],int);

void dispDetails(EMP [], int);

int main()

{

EMP e1;

EMP e[3];

int i;

/\*

e1.eId = 101;

e1.eSal = 10000.1;

strcpy(e1.eName,"Bhima");

strcpy(e1.eAddress,"KAR");

e1.eGender = 'M';

printf("\nEmployee Details are\n");

printf("\nID: %d",e1.eId);

printf("\nName: %s",e1.eName);

printf("\nGender: %c",e1.eGender);

printf("\nAddress: %s",e1.eAddress);

printf("\nSalary: %f",e1.eSal);

\*/

getDetails(e,3);

dispDetails(e,3);

printf("\n\n");

return 0;

}

void getDetails(EMP e[], int n)

{

int i;

printf("\nEnter Employee Details\n");

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

{

printf("\nName: ");

scanf("%s",e[i].eName);

printf("\nID: ");

scanf("%d",&e[i].eId);

printf("\nGender: ");

scanf(" ");

scanf("%c",&e[i].eGender);

printf("\nAddress: ");

scanf("%s",e[i].eAddress);

printf("\nSalary: ");

scanf("%f",&e[i].eSal);

}

}

void dispDetails(EMP e[], int n)

{

int i;

printf("\nEmployee Details are\n");

for(i=0;i<n;i++){

printf("\n=========================\n");

printf("\nID: %d",e[i].eId);

printf("\nName: %s",e[i].eName);

printf("\nGender: %c",e[i].eGender);

printf("\nAddress: %s",e[i].eAddress);

printf("\nSalary: %f",e[i].eSal);

printf("\n=========================\n");

}

}