**DAY 12**

**ARRAYS:**

Int a[2][3]

for (i=0; i<2 ; i++) {

for(j=0 , j<3, j++)

scanf (“

START

Read n=2

Set i=0

A1

i<n

i = i+1

Print i

**STRING CONCATINATION:**

#include <stdio.h>

#include<string.h>

int main()

{

int c1,c2,c3;

char res[40]={'\0',};

char temp[10] = {'\10',};

c1=2;

c2=1;

c3=0;

// 2\* 1=2

// c1 x c2 =c3

sprintf(temp,"%d", c1);

strcpy(res,temp);

printf("\nres = %s\n",res);

return 0;

} // Output🡪 res =2

**Code 2: To Print The 2 Table Multiplication?**

**Ans:**

{

int c1,c2,c3;

char res[40]={'\0',};

char temp[10] = {'\0',};

char temp1[]={'2'};

printf("\nSizeof temp1=%d value=%s", sizeof(temp1),temp);

c1=2;

c2=1;

c3=0;

// 2\* 1=2

// c1 x c2 =c3

while(c2<11){

c3=c1\*c2;

sprintf(temp,"%d x %d = %d", c1 , c2, c3);

strcpy(res,temp);

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

c2++;

return 0;

**STRINGS**

* Strings are used for storing text/characters.
* Str len
* Str comp
* Str cat
* Str copy
* Str str
* Str char
* Str tok

1. **Strlen 🡪 int strlen(obj);**

Printf(“%d”, strlen(name”));

1. **Strcmp**

<0 (s1 <s2)

==0 (s1 ==s2)

>0 (s1> s2)

If (strcmp(s1,s2 ==0)

* Comparison Of First name And Last Name:

If (strcmp( e1. Fname,”Anjali”)==0) &&

If (strcmp( e1. Lname,”Boyapalli”)==0)

🡪 if ((exp1)&& (exp2))

If (e1.id ==101) && ( str

1. **Strcat(s1,s2) --- String Concatination**

S1 is appended with s2

S1= Anjali, s2= Boypalli

Strcat (s1,s2)=>

Result 🡪 s1 is appended by s2

New valuesof s1 and s2 are

S1= AnjaliBoyapalli, s2= Boyapalli

* Digits is not return in string Hence it cannot be used in Nesting

🡺**POINTERS**

Char s1[] = “Anjali,Anitha,Gayatri”; (Array Of Characters)

Char \*ptr = NULL; (Pointers point the base address)

Ptr = s1; (ptr is pointing to s1 i.e ptr =2000)

Printf( “%c”, \*ptr) // Output = A (becoz it is in the char)

For (i=0, i<strlen(s1) ,i++)

{

If (\*ptr =’,’)

Break; (Coming Out Of Loop)

Ptr ++;

} // output of ptr =2006 at index value =6

🡺

for (i=0; ptr!=NULL; i++, ptr ++) -- Until ptr != to null it will loop)

{

If (\*ptr =’,’)

Break;

} // Output =11

🡪

For ( ; ptr !=NULL ; ptr++)

{ If (\*ptr =’,’)

Break;

}

A screen shot of a computer program

Description automatically generated

#include <stdio.h>

int main()

{

char Lines[] = "413|Anjalli|F|1310|10000";

char Names[10][20];

int row=0,col=0;

char \*ptr=NULL;

int flag = 0;

ptr = Lines;

while(\*ptr!='\0')

{

putchar(\*ptr);

ptr++;

}

ptr = Lines;

row=0;

col=0;

while(\*ptr != '\0')

{

if(\*ptr == '|')

break;

Names[row][col]=\*ptr;

ptr++;

col++;

}

Names[row][col] = '\0';

printf("\n\n");

putchar(\*ptr);

printf("\n\n");

puts(Names[row]);

while(1){ //if while (flag !=0)🡺 it ends with only

ptr++;

row++;

col=0;

if(\*ptr == '\0')

{

flag = 1;

break;

}

while(\*ptr != '\0')

{

if(\*ptr == '|')

break;

Names[row][col]=\*ptr;

ptr++;

col++;

}

Names[row][col] = '\0';

// printf("\n\n");

// putchar(\*ptr);

printf("\n\n");

puts(Names[row]);

}

/\*

ptr++;

row++;

col=0;

while(\*ptr != '\0')

{

if(\*ptr == '|')

break;

Names[row][col]=\*ptr;

ptr++;

col++;

}

Names[row][col] = '\0';

printf("\n\n");

putchar(\*ptr);

printf("\n\n");

puts(Names[row]);

\*/

printf("\n\n");

return 0;

}

**STRUCTERS**

It is the user defined data type

We want to create multiple data type of its type

Ex: Laptop is called data type(hp,dell,lenova)

Binding Of All diff data types makes a structure

Struct tagName{

Members /propetirs of the structure

};

**NESTED STRUCTURE :**struct with in a struct

**Syntax:**

Struct tagName1{

Members /propetirs of the structure

Sturct tagName2{

}

};

Typedef struct Employeee

{

Int id;

Int sal;

Int phno;

Char name[20];

Char Gender;

}EMP;

**FUNCTIONS:**

To perform Particular Task(add,sub)

**Syntax:**

Rdt fName(args)

{

Return rtd;

}

Display fails when the file is empty

Return Success is 1

**DYNAMIC MEMORY ALLOCATION:**

Is Used For Space Optimization

Pointer is necessary to build the address in dynamic allocation.

**Code:**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef struct Employeee

{

int id;

int sal;

int phno;

char Name[20];

char Gender;

}EMP;

int printEmp(EMP \*);

int getEmp(EMP \*);

int main()

{

EMP e1;

EMP \*e=NULL;

e = &e1;

scanf("%d%d%d",&e1.id,&e1.sal,&e1.phno);

scanf("%s",e1.Name);

getchar();

scanf("%c",&e1.Gender);

printf("\nID: %d", e->id);

printf("\nName: %s", e->Name);

printf("\nGender: %c", e->Gender);

printf("\nphno: %d", e->phno);

printf("\nsal: %d", e->sal);

printf("\n\n");

return 0;

}

* In Generic(void type=contains all type of data) we do type Casting.
* Free() :Invalid Pointer

Aborted (core dumped)