**DAY-7 (4-11)**

* **Managing input output operations**: reading a single character can be done by using the function getchar, this can also be done by scanf().

Eg: printf(), scanf(), getchar(), gets(), putchar(), puts()

* Two types of formatted I/O statements:

1. Formatted I/O – this enables the user specify the type of the data or the way it is read or written. Eg: scanf(), printf()
2. Unformatted I/O – it does not specify the type of the data or the way it is read or written. Eg: getchar(), gets(), putchar(), puts() etc

* %lf – for double

%u – usigned value

* Int num = 12345;

Scanf(“%3d”, &num); - o/p 123

3 is the field width of the input number

Here 45 will be in the temporary buffer

* Printf(“%05d”, 678);

0         0          6          7          8 – o/p

* Printf(“% 5d”, 678);

                      6          7          8 – o/p

If you give more than 5 digits of number, we get the output printed.

* To clear/flush out the buffer – fflush(stdin) function is used ( it will not work in the linux). It should be used before the scanf statement.

To clear buffer we can give scanf(“ “) – it is useful for the single character constant

To clear buffer numeric constant, we use

* Space or new line character is an end of string in scanf

Eg: name = pravallika c – o/p: Pravallika

To overcome this we give scanf(“%[^/n]s”, name); (it means it will scan all characters until new line)

* All format specifiers in scanf should not have any spaces it will slightly affect the output

Eg : scanf(“%d %d”,&n1,&n2); - not a problem

       scanf(“%d%d”, &n1, &n2); - not a problem

       scanf(“ %d%d”,&n1,&n2); - not a problem only for integers if the immediate one is character it will take as 3 inputs

       scanf(“%d %d ”,&n1,&n2); - slight affect

* **\t, \n should not be used in scanf (**avoid unprintable characters)

We can use anything in the printf

* sprintf is used to convert integer to string in linux

itoa in unix environment (p4.c)

* atoi possible in linux used to change a string to integer (only possible for numerical string)
* sprintf – to convert the integer into string. Should include library function when using sprintf

Eg: #include <stdio.h>

 2 #include <stdlib.h>

 3

 4 int main()

 5 {

 6    int i,j;

 7    char w[5] = "2002";

 8    char buff[1024];

 9    char buff1[1024];

10

11    int res=0;

12    scanf("%d%d",&i,&j);

13    printf("\ni=%d\tj=%d",i,j);

14    sprintf(buff,"\ni=%d\tj=%d",i,j);

15    printf("\nBuffer value: %s",buff);

16    sprintf(buff1,"%d",i);

17    printf("\nBuff1: %s",buff1);

18    res = atoi("2002")+2;

19    printf("\nResult = %d",res);

20    printf("\n\n");

21    return 0;

22 }

**O/p:** 10 20

i=10    j=20

Buffer value:

i=10    j=20

Buff1: 10

Result = 2004

* **decision making and branching**
* when we have more than one conditions and to test/ check, we use branching
* if statement, if else, nested if else, switch statement, else if ladder
* nested if:

if(cond1)

{

          If(cond2)

          {

          }

          Else

          {

          }

}

Else {

If(cond3)

          {

          }

          Else

          {

          }

}