**DAY-7**

**Managing Input and output operations**:

**Reading a character:**

* Reading a single character can be done by using the function getchar(). And can also be done by scanf()

**syntax**: ch = getchar()

**Input/output statements:**

* To perform basic i/o functions C provides the library of functions . This library is called as stdio.h

e.g.: scanf(),printf(),getchar(),getch(),gets(), puts(), putchar()

there are 2 type:

**1. Formatted i/o statement**

It enable the user to specify the data and the way in which it should be read in or written out

eg: scanf(), printf()

**2. Unformatted i/o statement**

This do not specify the type of data and the way in which it should be read in or written out

eg:getchar(),getch(),gets(), puts(), putchar()etc

**scanf() syntax:**

scanf("control string",address\_list);

where, control string is a sequence of one or more character groups.

**Format specifiers:**

%c --- read single character

%d --- read decimal integer

%f --- read a float value

%u --- read an unsigned value

**Integer Input:**

eg: num = 386;

scanf("%3d",&num);

3 is the field width of the input number

Eg:scanf("%05d",685);--- if the value has only 3 digits rest will be stored as 0.

gcc <file name>

./a.out

#include <stdio.h>

int main(){

int i;

float f;

char ch;

char str1[20];

double d;

printf("\nEnter the proper values\n");

printf("\nEnter the integer value:");

scanf("%d",&i);

printf("\ninteger value: %05d\n",i);

return 0;

}

fflush stdin

**ERROR**: Stack smashing detected aborted(core dump) ---- to clear up the unreserved space we don't have the permissions.

signal interruption has occurred.

**address of variables:**

i -- 472

f --- 476

ch -- 471

str1 -- 488

d -- 480

these are known as base address of any variable.

sprintf is used to store the buffer value.

#include <stdio.h>

int main(){

int i,j,k;

char buff[100] = "10 20 30";

sscanf(buff,"%d %d %d",&i,&j,&k);

printf("\ni=%d\tj=%d\tk=%d\n",i,j,k);

return 0;

}

**unformatted io**

getc ,putc -- read/write single char -> stream

getchar, putchar -- read/write single char -> std i/o ---- reads only single character and doesn't check for any errors

gets ,puts --- read/write for a string -> stream

getch()---- DOS platform ---> not echo read char to the screen

#include <stdio.h>

int main(){

char ch;

printf("\nEnter a character:");

ch = getchar();

printf("\nread char:");

putchar(ch);

putchar('A');

putchar(65);

printf("\n\n");

return 0;

}

char ch;

ch = fgetc(stdin);

printf("\nRead character: ");

putc(ch,stdout);

printf("\n\n");

return 0;

STRING HANDELING:

**Decision making and Branching:**

**Branching**: to check the multiple conditions

\*\*\*\*boolean is not a datatype in c\*\*\*\*

**1.if**

**Syntax:**

if(cond)

{

if block of statements

}

eg:

#include<stdio.h>

int main(){

int age;

scanf("%d",&age);

if(age>=18)

{

printf("\nyou are eligible to vote");

}

printf("\nIndian citizen");

return 0;

}

**2.if else**

**syntax:**

if(cond)

{

if block of statements

}

Else if(cond 2)

{

}

else if(cond 3)

{

}

Else

{

}

Eg;

#include<stdio.h>

int main(){

char clrcode;

clrcode = getchar();

if(clrcode == 'B')

printf("\nBlack\n");

else if(clrcode == 'N')

printf("\nBrown\n");

else if (clrcode == 'R')

printf("\nRed\n");

else if(clrcode == 'O')

printf("\nOrange\n");

else

printf("\nEnter the correct color code\n");

printf("\nprogram ended\n");

return 0;

}

**3.nested if else**

**Syntax**:

if(cond 1)

{

if(cond2)

{

}

else

{

}

}

else

{

if(cond3)

{

}

else

{

}

**4.switch ----** values will be in numeric or ASCII due to binary search.

Switch(cond)

{

Case 1:

-----

break;

Case 2:

-----

break;

default:

-----

break;

}

e.g.:

#include <stdio.h>

int main(){

char clrcode;

clrcode = getc(stdin);

switch(clrcode)

{

case 'O':

printf("Orange");

break;

case 'B':

printf("Black");

break;

case 'N':

printf("Brown");

break;

default:

printf("\nEnter the correct color code\n");

break;

}

printf("\nProgram Ended\n");

return 0;

}