**DAY 08**

**chapter 5**

**Decision making & Branching statement**

if

else

nested if

* else if ladder the condition is contitued until it meets the condition

if(condition1)

{

}

else if(condition2)

{

}

else if(condition3)

{

}

else

{

}

* this is used in resistance values calculation.

grading of marks,time table,planning trip.

* switch case is faster than else if ladder
* else if ladder

whenever alphanumeric ,float values and strings are to be tested this can be used

**SWITCH**

switch(cond)

{

case 1:

---

break;

case 2:

-------

break;

default :

-------

break;

}

* **case values**

either it should be a numeric,single character constants,ascii values

but it cannot be a string,float because it undergoes search condition

it forms a table of unique value .

process exited normally -gdb error

b 31

r

n

l

* **loop**

1.initialize variable

2.cond check

3. sts to be executed

4.counter

* 3types:

1.do while

2.exit controlled

d0{

sts

}

while(con);

if the if loop is going infinetly the scanf is not implemented properly

exit(success)

#define success 0

#define failure -1

macro definition

#define True 1

#define false 0

for(initial section;cond section;counter)

{

sts;

}

i=0;

j=0;

for(;;)

{

sts;

if((i>10)&&(j>10))

break;

i++;

j+=2;

}

**nested loop**

for(i=0;i<10;i++);

{

for (j<10;j=0:j++);

printf("i=%d j=%d",i,j);

**}**

for loop one or more variables can be initialized

for loop is more faster than while loop

for loop can have more than one counter

in a single line initialization ,conditions,counter can be written.

one or more section can be omitted

**can for loop converted to exit control loop instead of entry control?**

#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 colour code");

printf("\nProgram Ended\n");

return 0;

}