**DAY -8 (5-11-2024)**

* **Decision making and branching:**

1. If
2. If else
3. Nested if else
4. Switch statements

* Syntax:

switch(cond)

{

case 1:

…..

break;

Case2:

..

break;

default:

…

Break;

}

* In switch case we use for single character, single numeric value (ASIIC VALUES).
* It directly jumps to that particular case it doesn’t execute other case statements.
* Applications: menu based
* If else can have the switch and verse vice is also applicable.

#include<stdio.h>

int main()

{

char clrCode;

clrCode=getc(stdin);

switch(clrCode)

{

case 'O':

printf("orange");

break;

case 'B':

printf("black");

break;

default:

printf("correct code");

break;

}

printf("\nended\n");

return 0;

}

* If duplicate value is given in case then it throws an error.
* case ‘b’ and case ‘b’ is not acceptable(duplicates)
* case 66 and case ‘b’ is also not acceptable.(duplicates)
* Space complexity can be reduced by remove the clrCode character and can write as switch(getchar(stdin))

1. Else if ladder

* Multiple test cases are tested at the root level
* Syntax:

If(cond1)

{

}

else if(cond2)

{

}

else if(cond3)

{

}

else

{

}

* Application: to calculator the value of the resistor, Grading system
* Switch case is faster than else if ladder
* For testing alphanumeric, string, float in else if ladder.

* “r” is used to run in debugging mode
* Error: Exited normally in debugging mode.
* **Decision making and looping:**
* Looping : used for repetitive tasks

1. Initialization variable
2. Condition check
3. Statements to be executed
4. Counter

* 3 different types of loops:

1. Do while
2. While
3. For

Two types of loops:

1. Entry controlled: check the condition at the entry itself and then it executed the statements
2. Exit controlled: it executed the statements and then check the condition at the exit

Eg: do while

Do- while :

* Syntax:

do{

}

While(cond);

* It have the semicolon at the end
* Why is that semicolon at the end of the while loop

A computer screen shot of a black screen

Description automatically generated

While loop :

* Syntax:

while(cond)

{

}

* It checks the condition if the condition is true then the statements are executed if the condition met wrong then it comes out of loop
* While(1) is a infinte loop-we here use numeric constant

1 for the True value

Instead of 1 we can use the macros

#define TRUE 1

#define FALSE 0

* Scanf is does not work properly then for the scanf we have to handle the error of scanf statements
* exit(SUCCESS) is used to exit the code if it is infinite loop.we have to define the macros for this like

#define SUCCESS 0

#define FAILURE -1

In last mention return SUCCESS instead of return 0

We have to mention #include<stdlib.h>

If we use exit it means we killing the program or not exit the program normally if you want to exit the program normally we have to use the flags.

**For loop:**

* Syntax

For(initial section; cond section; counter)

{

}

* In for loop we can initialization one or more variables and also have the more than one counter and also more than one condition checking
* Eg : for(i=0,j=0;(i<=3)&&(j<=6); i++,j+=2)
* We can omit the sections like

i=0,j=0;

for(;(i<=3)&&(j<=6); i++,j+=2)

for(;; )

{

If (i<=3)&&(j<=6){

}

i++;

j+=2;

}

If here we can give only give the one condition also to execute both the I and j values.