1. Following code returns false

#include<stdio.h>

void main(){

float x = 0.1;

if(x == 0.1){

printf("true");

}else{

printf("false");

}

}

Lets check the datatype of x and 0.1

printf("%d\n",sizeof(x)); //4 bytes

printf("%d\n",sizeof(0.1)); //8 bytes - treated as long so **x** is not equal to 0.1

Fix : **use 0.1f instead of 0.1**

if(x == 0.1f){

printf("true\n");

}else{

printf("false\n");

}

2.To print ASCII value of given character

#include <stdio.h>

int main()

{

char c = 'a';

int i = c;

printf("the ascii value of %c is %d ",c,i);

}

3.Convert ASCII to char

#include <stdio.h>

int main()

{

int i = 116;

char c = i;

printf("%c\n",c);

}

//output : ASCII value of **t** is 116

4.**%d and %i** both format specifiers can be used to display int datatype

#include <stdio.h>

int main()

{

int i = 10;

printf("%d\n",i);

printf("%i\n",i);

}

5.prints fixed number of characters after decimal point

#include <stdio.h>

int main ()

{

float x = 1.5;

printf("%f\n",x);

printf("%.2f",x);

//%.2f - prints floating point , 2 characters after decimal point

return 0;

}

6. Printing percentage symbol

printf("%");//doesn't prints %

printf("\n");

printf("%%");//prints % once

7.declaring Symbolic Constants(pi, gravity, speed of light etc , custom constants )

A **#define** line defines a symbolic name or symbolic constant to be a particular string of characters

Syntax : #define name replacementText/number

#include <stdio.h>

#include <stdbool.h>

#define pi 3.14

#define schoolname "vidya\_madira"

#define holiday true

int main ()

{

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

printf("%f\n",pi);

printf("%d",holiday);

return 0;

}

8.Increment operator

#include <stdio.h>

int main ()

{

int x = 5;

int y = ++x;

printf("y = %d\n",y);

printf("x = %d\n",x);

int a = 5;

int b = a++;

printf("b = %d\n",b);

printf("a = %d\n",a);

return 0;

}

**9.**call by value does not works for arrays.

Reason : When the name of an array is used as an argument, **the value passed to the function is the location or address of the beginning of the array-there is no copying of array elements.**

**//** arrays can be passed using reference not by value

#include <stdio.h>

void test(int a[]){

a[0] = 10;

printf("a[0] is %d\n",a[0]);

}

int main ()

{

int a[5] = {5,10,15};

test(a);

printf("after test = %d",a[0]);

return 0;

}