Accepting Multiple Values From The User
=================================

Qn. WAP to ask the user to input his grade , roll number and percentage
and then display these values back on screen


```c
#include <stdio.h>
#include <conio.h>
void main()
{
    int r;
    char g;
    float p;
    clrscr();
    printf("Enter your grade:");
    scanf("%c",&g);
    printf("Enter your roll no:");
    scanf("%d",&r);
    printf("Enter your per:");
    scanf("%f",&p);
    printf("Your grade is %c\nYour roll no is %d\nYour per is %f",g,r,p);
    getch();
}
```

Qn Why the code is showing 6 digits after the decimal point ?

Ans: In C lang , the digits displayed after the decimal point in float values is called as **precision** and this precision by default is upto 6 places.

Qn. Can we change this behaviour ?

Ans: Yes we can do that but for this we will have to write printf() in a special way:

This tells the compiler that how many digits we want after decimal point

Syntax:
======
    printf("message %.<number>f",var....);

Ex:
===
    printf("Your percentage is %.2f",p);

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    int r;
    char g;
    float p;
    clrscr();
    printf("Enter your grade:");
    scanf("%c",&g);
    printf("Enter your roll no:");
    scanf("%d",&r);
    printf("Enter your per:");
    scanf("%f",&p);
    printf("Your grade is %c\nYour roll no is %d\nYour per is %.2f",g,r,p);
    getch();
}
```

formatting the output

Using A Single scanf() For Scanning Multiple Values
==

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    int r;
    char g;
    float p;
    clrscr();
    printf("Enter your grade , roll no and per:");
    scanf("%c %d %f",&g,&r,&p);
    printf("Your grade is %c\nYour roll no is %d\nYour per is %.2f",g,r,p);
    getch();
}
```
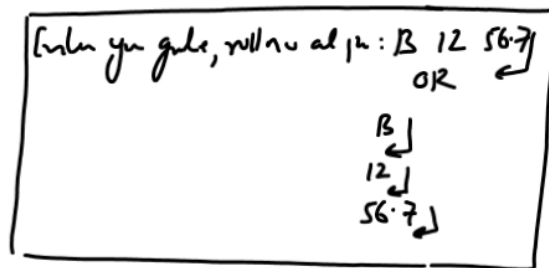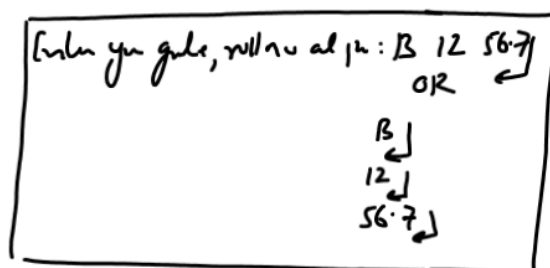

Enter your grade, roll no and per : B 12 56.7

OR

B
12
56.7

int r; X
float r;

int r; ✓
float R; ✓ OK, but not
recommend

X int char;
X float int;

int Char; ✓ ok but not
recommend

2. Variable names can contains alphabets as well as digits but they must begin with an alphabet

✓ int r1;

✓ int r12;

X int 1r;

X int 1;

✓ int R;

✓ int B;

✓ int C1;