Q. How to find square and cube of a number using macros?

#define preprocessor directive in C program.

Logic to find square and cube of a number using macro.

Till now we have covered basics of macro how to define, undefine and redefine a macro in C programming.

In this class I will explain how to find square and sum of two numbers using macro, using #define preprocessor directive in C program.

Program will be:

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■] MACRO1.C 1=[↕]
/* C program to find square and cube of a number using macro */
#include <stdio.h>
// Define macro to find square and cube
#define SQUARE(x) (x * x)
#define CUBE(x) (x * x * x)
void main()
{
    int num;
     // Input a number from user
    printf("Enter any number to find square and cube: ");
    scanf("%d", &num);
     // Calculate and print square
    printf("SQUARE(%d) = %d\n", num, SQUARE(num));
     // Calculate and print cube
    printf("CUBE(%d) = %d\n", num, CUBE(num));
    getch();
}
11:65
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

Output will be as

```
C:\TC\BIN>tc.exe
Enter any number to find square and cube: 9
SQUARE(9) = 81
CUBE(9) = 729
```

All types of Basic Data Input through Programming in C.

Points to note:

- `\n` is an escape sequence character used to print new line (move to next line).
- The `getchar()` function reads single character and stores to some character variable.
- During a character input suppose we input `C` and then enter which is also considered as a character.
- Internally, `getchar()` reads and stores `C` character to `charVal` and tries to store the enter character to `uCharVal`.
- This is because I have used an extra `getchar()` to eliminate the enter character.

Example to do:-

```
/* C program to demonstrate input output of primitive data types */

 #include <stdio.h>

void main()

{

/*

* Declare all primitive and derived types

*/

   char charVal;

    unsigned char uCharVal;

    short shortVal;

    unsigned short uShortVal;

    int intVal;

    unsigned int uIntVal;
```

```
long longVal;

unsigned long uLongVal;

long long longLongVal;

unsigned long long uLongLongVal;

float floatVal;

double doubleVal;

long double longDoubleVal;

/*

 * Read input in each type

 */

printf("Enter a character: ");

charVal = getchar();

getchar(); // <-- Dummy getchar() to capture enter

printf("Enter another character: ");

uCharVal = getchar();

getchar(); // <-- Dummy getchar() to capture enter

printf("Enter a signed short value: ");

scanf("%hi", &shortVal);

printf("Enter an unsigned short value: ");

scanf("%hu", &uShortVal);

printf("Enter an signed integer value: ");

scanf("%d", &intVal);

printf("Enter an unsigned integer value: ");

scanf("%lu", &uIntVal);

printf("Enter a signed long value: ");

scanf("%ld", &longVal);

printf("Enter an unsigned long value: ");

scanf("%lu", &uLongVal);
```

```
printf("Enter a signed long long value: ");
 scanf("%lld", &longLongVal);
 printf("Enter an unsigned long long value: ");
 scanf("%llu", &uLongLongVal);
 printf("Enter a float value: ");
 scanf("%f", &floatVal);
 printf("Enter a double value: ");
 scanf("%lf", &doubleVal);
 printf("Enter a long double value: ");
 scanf("%Lf", &longDoubleVal);
 /*
  * Print the value of all variable
  */
 printf("\nYou entered character: '%c' \n", charVal);
 printf("You entered unsigned character: '%c' \n\n", uCharVal);

 printf("You entered signed short: %hi \n", shortVal);
 printf("You entered unsigned short: %hu \n\n", uShortVal);

 printf("You entered signed int: %d \n", intVal);
 printf("You entered unsigned int: %lu \n\n", uIntVal);

 printf("You entered signed long: %ld \n", longVal);
 printf("You entered unsigned long: %lu \n\n", uLongVal);

 printf("You entered signed long long: %lld \n", longLongVal);
 printf("You entered unsigned long long: %llu \n\n", uLongLongVal);
```

```
    printf("You entered float: %f \n", floatVal);

    printf("You entered double: %lf \n", doubleVal);

    printf("You entered long double: %Lf \n", longDoubleVal);

getch();

}
```

Output will be :

```
Enter a character: C
Enter another character: P
Enter a signed short value: -32768
Enter an unsigned short value: 65535
Enter an signed integer value: -2147483648
Enter an unsigned integer value: 4294967295
Enter a signed long value: -2147483648
Enter an unsigned long value: 4294967295
Enter a signed long long value: -9223372036854775808
Enter an unsigned long long value: 18446744073709551615
Enter a float value: 1.28766
Enter a double value: 10.915074
Enter a long double value: 100.12345

You entered character: 'C'
```

```
You entered unsigned character: 'P'

You entered signed short: -32768
You entered unsigned short: 65535

You entered signed int: -2147483648
You entered unsigned int: 4294967295

You entered signed long: -2147483648
You entered unsigned long: 4294967295

You entered signed long long: -9223372036854775808
You entered unsigned long long: 18446744073709551615

You entered float: 1.287660
```

```
You entered float: 1.287660
You entered double: 10.915074
You entered long double: 100.123450
```