S.No: 25	Exp. Name: ***Illustrate the use of register variables***	Date:2023-04-01

Aim:

Illustrate the use of register variables.

- You can use the **register** storage class when you want to store local variables within functions or blocks in CPU registers instead of RAM to have quick access to these variables. For example, "counters" are a good candidate to be stored in the register.
- The keyword **register** is used to declare a register storage class. The variables declared using register storage class has lifespan throughout the program.
- It is similar to the auto storage class. The variable is limited to the particular block. The only difference is that the variables declared using register storage class are stored inside CPU registers instead of a memory. Register has faster access than that of the main memory.
- The variables declared using register storage class has no default value. These variables are often declared at the beginning of a program.
- Accessing the address of the register variables results in an error.

Try it out

```
A statement like
int *ptr = &weight;
will result in an error like
int *ptr = &weight;
address of register variable 'weight' requested
```

Follow the instructions given in the comment lines to understand the working of register variables.

Source Code:

register.c

```
#include <stdio.h>
void main() {
register int weight; // Declare a register variable weight of type int.
   printf("The default weight value is: %d\n",weight);
   weight=65;
printf("The current weight value is: %d\n",weight);   // Add the line described above
to obtain the error.
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
The default weight value is: 1024482696