C library function - asctime()

Description

The C library function **char *asctime(const struct tm *timeptr)** returns a pointer to a string which represents the day and time of the structure **struct timeptr**.

Declaration

Following is the declaration for asctime() function.

```
char *asctime(const struct tm *timeptr)
```

Parameters

The **timeptr** is a pointer to tm structure that contains a calendar time broken down into its components as shown below −

```
struct tm {
   int tm_sec;         /* seconds,  range 0 to 59          */
   int tm_min;         /* minutes, range 0 to 59           */
   int tm_hour;        /* hours, range 0 to 23             */
   int tm_mday;        /* day of the month, range 1 to 31  */
   int tm_mon;         /* month, range 0 to 11             */
   int tm_year;        /* The number of years since 1900   */
   int tm_wday;        /* day of the week, range 0 to 6    */
   int tm_yday;        /* day in the year, range 0 to 365  */
   int tm_isdst;       /* daylight saving time             */
};
```

Return Value

This function returns a C string containing the date and time information in a human-readable format **Www Mmm dd hh:mm:ss yyyy**, where *Www* is the weekday, *Mmm* the month in letters, *dd* the day of the month, *hh:mm:ss* the time, and *yyyy* the year.

Example

The following example shows the usage of asctime() function.

Live Demo

```
#include <stdio.h>
#include <string.h>
#include <time.h>
```

```
int main () {
   struct tm t;

   t.tm_sec    = 10;
   t.tm_min    = 10;
   t.tm_hour   = 6;
   t.tm_mday   = 25;
   t.tm_mon    = 2;
   t.tm_year   = 89;
   t.tm_wday   = 6;

   puts(asctime(&t));

   return(0);
}
```

Let us compile and run the above program that will produce the following result −

```
Sat Mar 25 06:10:10 1989
```