

**Write structure of program to filter out non-posix compliant codes from user program
(OR) Write a program to print the version of UNIX system.**

```
#define _POSIX_SOURCE

#define _POSIX_C_SOURCE 199309L

#include <iostream.h>

#include <unistd.h>

int main()

{

#ifdef _POSIX_VERSION

cout << "System conforms to POSIX: " << _POSIX_VERSION << endl;

#else

cout << "_POSIX_VERSION is undefined\n";

#endif

return 0;

}
```

LAB-2-TERMWORK

PROGRAM TO PRINT THE FEATURE TEST MACROS

```
#define _POSIX_SOURCE

#define _POSIX_C_SOURCE 199309L
```

```
#include <iostream.h>

#include <unistd.h>

int main()
{
#ifdef _POSIX_JOB_CONTROL

    cout << "System supports job control\n";

#else

    cout << "System does not support job control\n";

#endif

#ifdef _POSIX_SAVED_IDS

    cout << "System supports saved set-UID and saved set-GID\n";

#else

    cout << "System does not support saved set-UID & GID\n";

#endif

#ifdef _POSIX_CHOWN_RESTRICTED

    cout << "chown restricted option is: " << _POSIX_CHOWN_RESTRICTED << endl;

#else

    cout << "System does not support system-wide chown_restricted option\n";

#endif

}
```

```
#endif
```

```
#ifdef _POSIX_NO_TRUNC
```

```
    cout << "Pathname truncation option is: " << _POSIX_NO_TRUNC << endl;
```

```
#else
```

```
    cout << "System does not support system-wide pathname truncation option\n";
```

```
#endif
```

```
#ifdef _POSIX_VDISABLE
```

```
    cout << "Disable character for terminal files is: " << _POSIX_VDISABLE << endl;
```

```
#else
```

```
    cout << "System does not support _POSIX_VDISABLE\n";
```

```
#endif
```

```
    return 0;
```

```
}
```