```
int i;
char **ptr;
extern char **environ;
for(ptr=environ; *ptr!=0; ptr++)
pf("%s\n", *ptr);
return 0;
}
```

o/p: $ gcc env.c
$ ./a.out

O/P Sir 9/1/23

LAB-12

① Write a pgm to emulate ln command.

```
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
int main(int argc, char *argv[])
{ if (argc < 3 || argc > 4 || (argc==4
        && strcmp(argv[1], "-s")))
  {
    printf("usage: ./a.out [-s] <arg_file>
              <new_link> \n");
    return 1;
  }
```

papergrid

Date: / /

```
if (argc==4)
{
    if((symlink(argv[2],
    argv[3]))==-1)
    pf("cannot create symbolic link
    else
    printf("symbolic link created\n");
}
else
{ if((link(argv[1],argv[2]))==-
  pf("cannot create hardlink\n");
  else
  { pf("hard link created\n");
}
return 0;
}
```

O/p:-

$ ./a.out c11.c hema.
Hard link created.

$ ./a.out -s c11.c sh.
Symbolic link created.

2] write a pgm · POSIX compliant pgm that prints POSIX defined configuration.
options supported on any given sys using feature test macros.

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#define include <stdio.h>
#define include <unistd.h>
int main()
{
  #ifdef _POSIX_JOB_CONTROL
  pf("System supports job control\n");
  #else
  pf("System does not support job control\n");
  #endif
  #ifdef _POSIX_SAVED_IDS
  pf("System supports saved set UID
      and saved set-GID\n");
  #endif
  #ifdef _POSIX_CHOWN_RESTRICTED
  pf("chown restricted option is %d\n",
      _POSIX_CHOWN_RESTRICTED);
  #else
  pf("Sys does not support chown
      restricted option\n");
```

Date: / /

```
#include <stdio.h>
int main (int argc, char* argv[])
{ int fd, qt;
  char buf[256];
  if (argc != 2 && argc != 3)
  {
    pf("usage %s <file> [<args>]\n"
                            , argv[0]);
    Deturn 0;
  }
  mkfifo(argv[1], S_IFIFO|S_IRWX(|
                       S_IRWXO);
  if (argc == 2)
  {
    fd = open(argv[1], O_RDONLY|
                        O_NONBLOCK);
    while( read(fd, buf, sizeof(buf)),
           (!gt, 0)
      pf("%s\n", buf);
  }
  else {
       fd = open(argv[1], O_WRONLY);
       write(fd, argv[2], strlen(argv[2]));
  }
  closed(fd);
```

#end if

return 0;

}

o/p $ gcc c12.c
$ ./a.out

System supports job control.
System supports saved set UID and saved set GID.
Shown restricted option is 0
pathname trunc option is 1
Disable character per terminal is 0