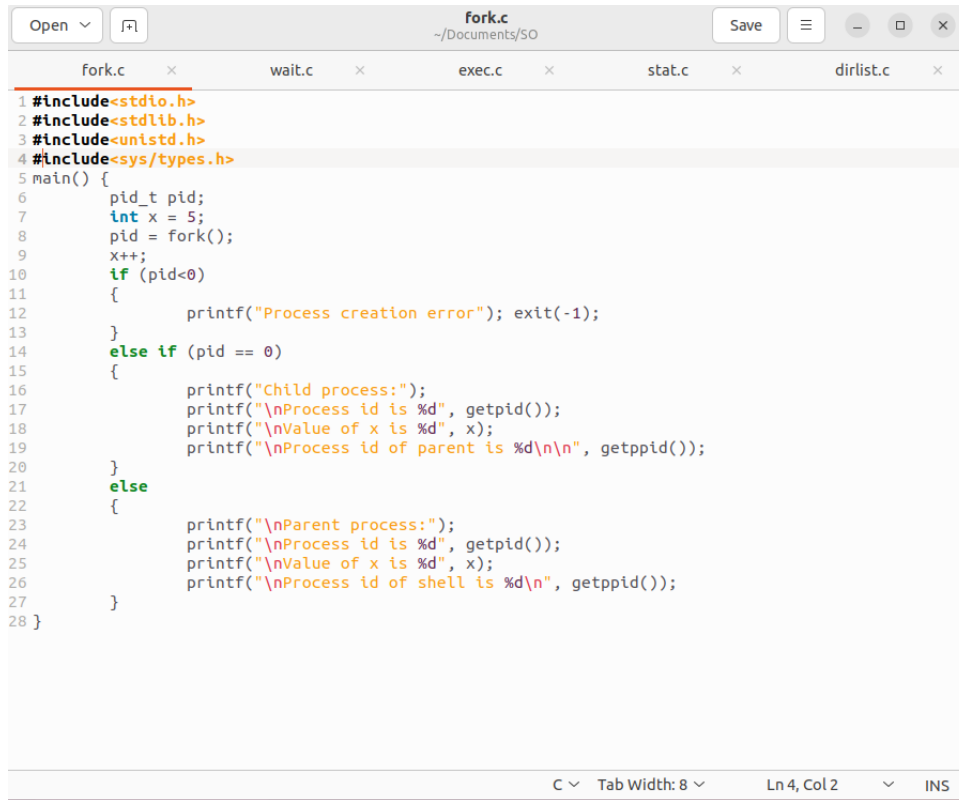


Nama = Maya Nurkhayati
NIM = L200210220
Kelas = E

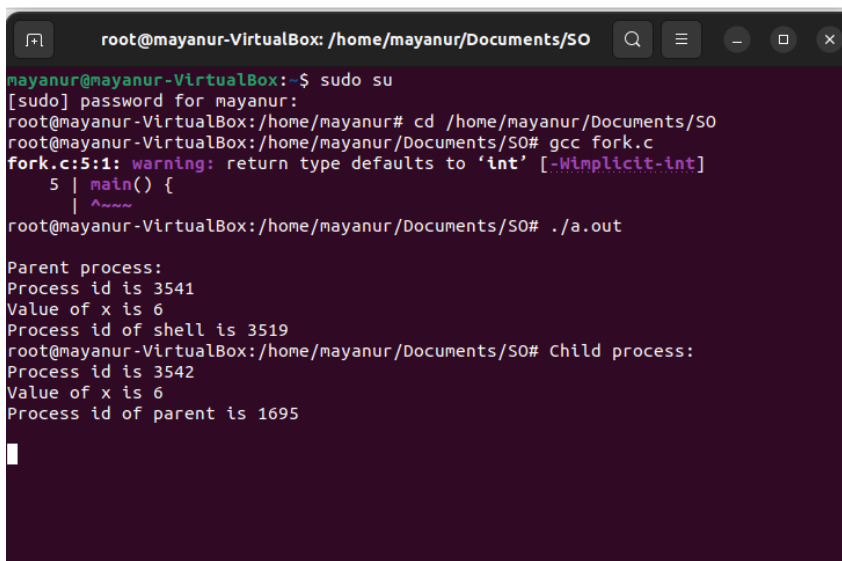
MODUL 8

1. FORK.C



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/types.h>
5 main() {
6     pid_t pid;
7     int x = 5;
8     pid = fork();
9     x++;
10    if (pid<0)
11    {
12        printf("Process creation error"); exit(-1);
13    }
14    else if (pid == 0)
15    {
16        printf("Child process:");
17        printf("\nProcess id is %d", getpid());
18        printf("\nValue of x is %d", x);
19        printf("\nProcess id of parent is %d\n", getppid());
20    }
21    else
22    {
23        printf("\nParent process:");
24        printf("\nProcess id is %d", getpid());
25        printf("\nValue of x is %d", x);
26        printf("\nProcess id of shell is %d\n", getppid());
27    }
28 }
```

Output



```
root@mayanur-VirtualBox: /home/mayanur/Documents/SO
mayanur@mayanur-VirtualBox:~$ sudo su
[sudo] password for mayanur:
root@mayanur-VirtualBox:/home/mayanur# cd /home/mayanur/Documents/SO
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# gcc fork.c
fork.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
    5 | main() {
      | ^~~~~
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# ./a.out

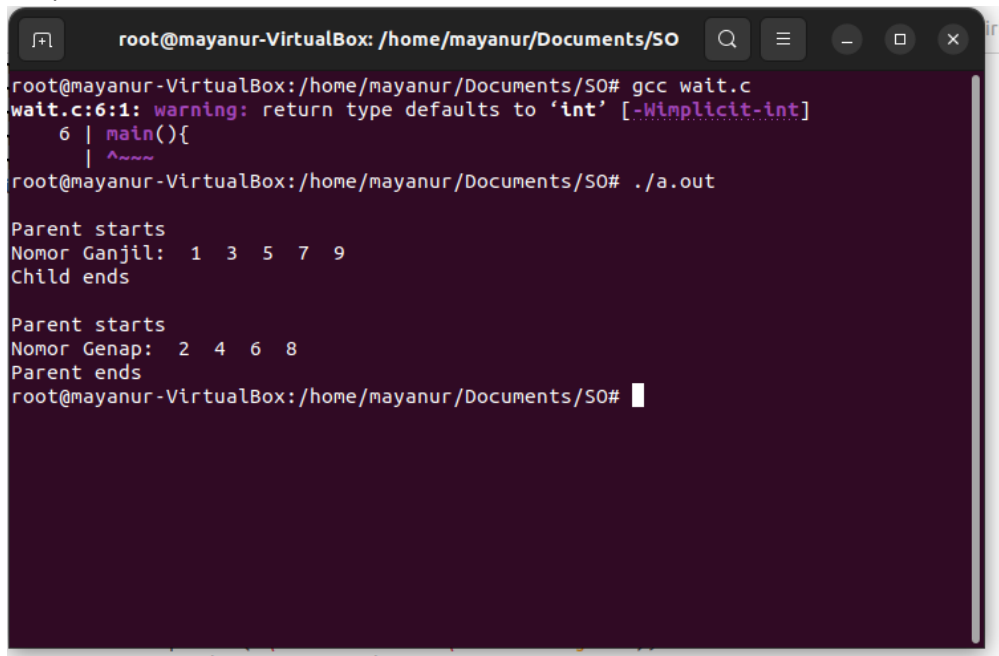
Parent process:
Process id is 3541
Value of x is 6
Process id of shell is 3519
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# Child process:
Process id is 3542
Value of x is 6
Process id of parent is 1695
```

2. WAIT.C



```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<sys/types.h>
5 #include<sys/wait.h>
6 main(){
7     int i, status;
8     pid_t pid;
9     pid = fork();
10
11     if(pid,0){
12         printf("\nPembuatan proses gagal\n");
13         exit(-1);
14     }
15     else if (pid>0)
16     {
17         wait(NULL);
18         printf("\nParent starts\nNomor Ganap:");
19         for(i=2;i<10;i+=2)
20             printf("%3d",i);
21         printf("\nParent ends\n");
22     }
23     else if(pid==0)
24     {
25         printf("\nParent starts\nNomor Ganjil:");
26         for(i=1;i<10;i+=2)
27             printf("%3d",i);
28         printf("\nChild ends\n");
29     }
30 }
```

Output



```
root@mayanur-VirtualBox: /home/mayanur/Documents/SO
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# gcc wait.c
wait.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
  6 | main(){
    | ^~~~~
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# ./a.out

Parent starts
Nomor Ganjil:  1  3  5  7  9
Child ends

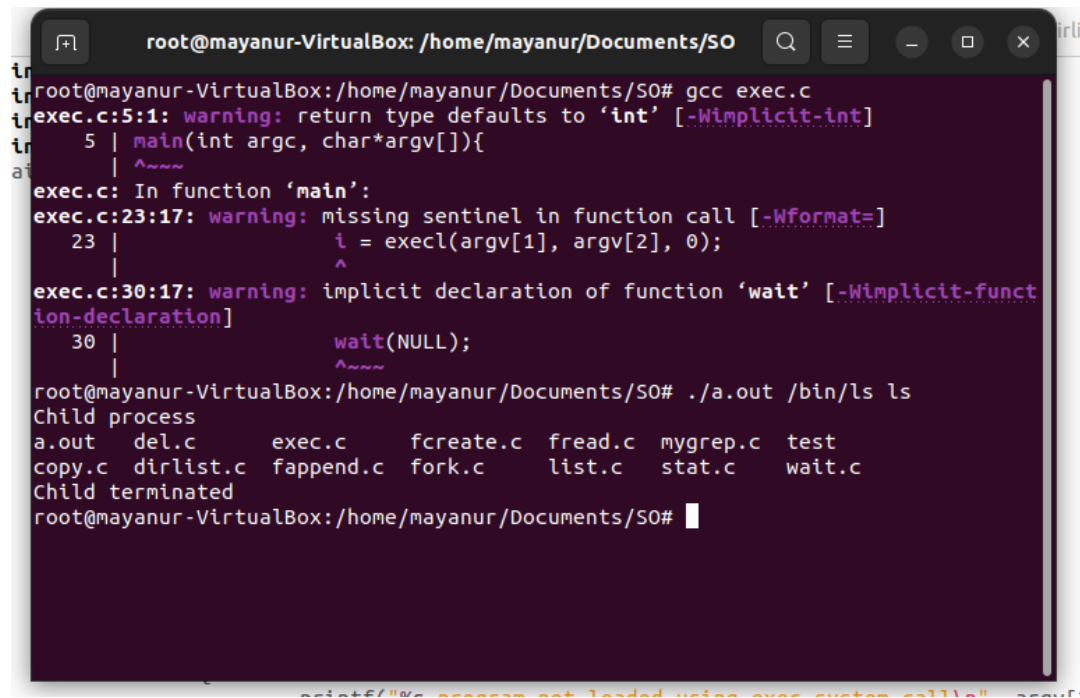
Parent starts
Nomor Genap:  2  4  6  8
Parent ends
root@mayanur-VirtualBox:/home/mayanur/Documents/SO#
```

3. EXEC.C



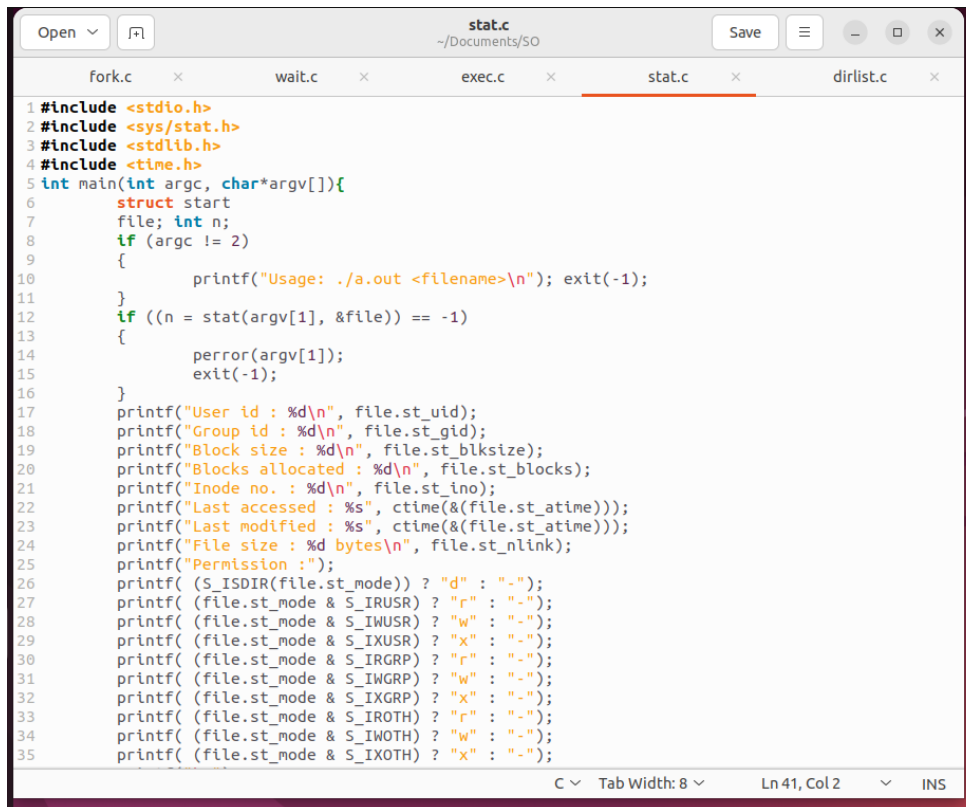
```
1 #include<stdio.h>
2 #include<sys/types.h>
3 #include<unistd.h>
4 #include<stdlib.h>
5 main(int argc, char*argv[]){
6
7     pid_t pid;
8     int i;
9
10    if (argc !=3)
11    {
12        printf("\nInsufficient arguments to load program");
13        printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
14    }
15
16    switch(pid = fork())
17    {
18    case -1:
19        printf("Fork failed");
20        exit(-1);
21    case 0:
22        printf("Child process\n");
23        i = execl(argv[1], argv[2], 0);
24        if (i<0)
25        {
26            printf("%s program not loaded using exec system call\n", argv[2]);
27            exit(-1);
28        }
29    default:
30        wait(NULL);
31        printf("Child terminated\n");
32        exit(0);
33    }
34 }
```

Output



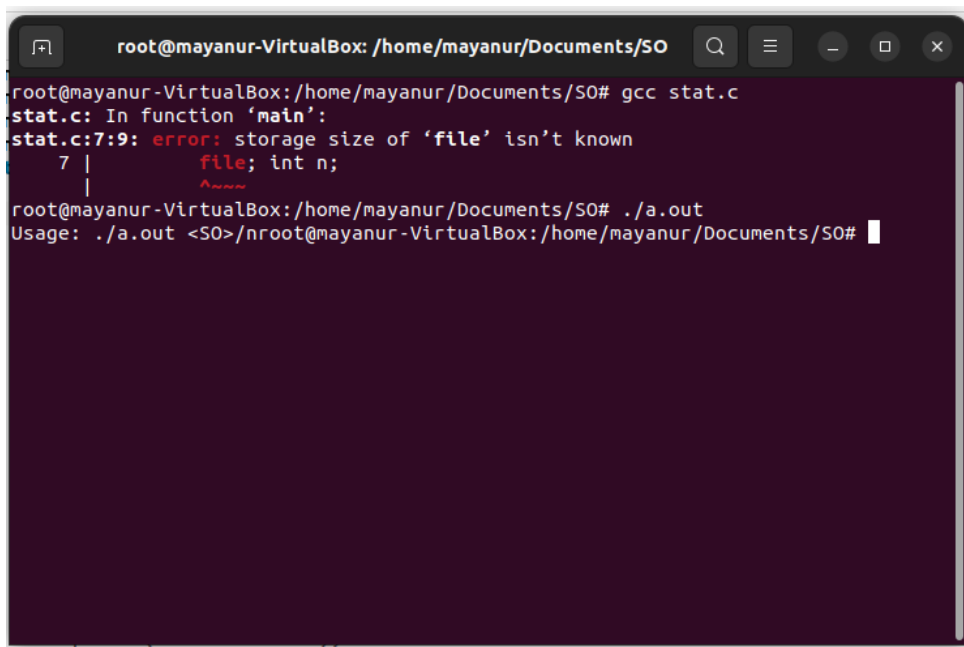
```
root@mayanur-VirtualBox: /home/mayanur/Documents/SO
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
5 | main(int argc, char*argv[]){
  | ^~~~~
exec.c: In function 'main':
exec.c:23:17: warning: missing sentinel in function call [-Wformat=]
23 |         i = execl(argv[1], argv[2], 0);
    |         ^
exec.c:30:17: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
30 |         wait(NULL);
    |         ^~~~~
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# ./a.out /bin/ls ls
Child process
a.out del.c      exec.c      fcreate.c  fread.c   mygrep.c  test
copy.c dirlist.c fappend.c fork.c    list.c    stat.c    wait.c
Child terminated
root@mayanur-VirtualBox:/home/mayanur/Documents/SO#
```

4. STAT.C



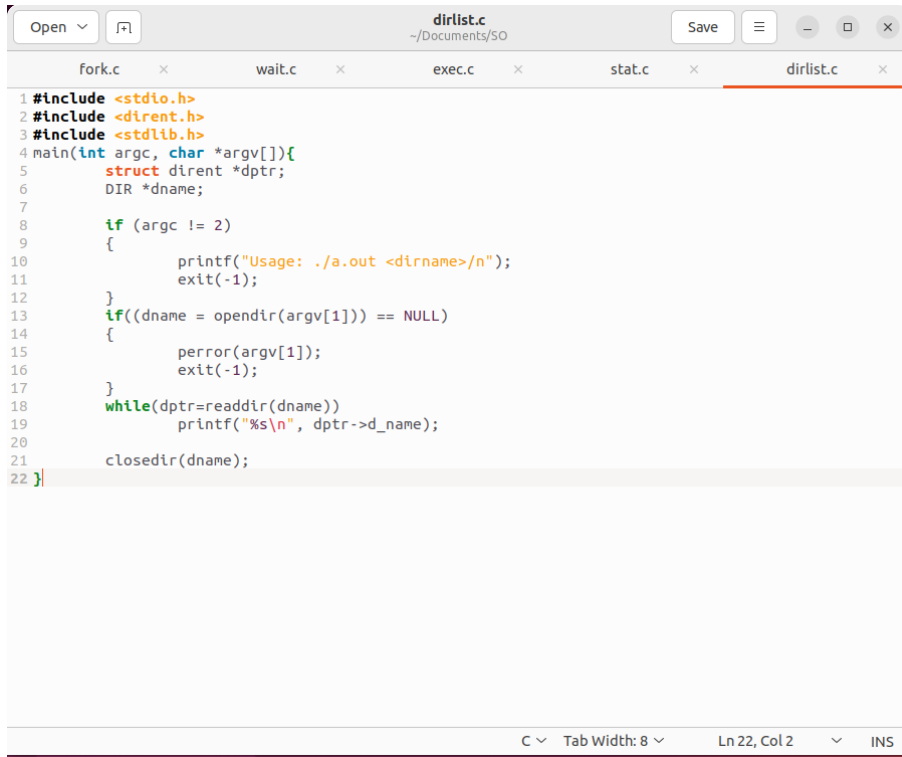
```
1 #include <stdio.h>
2 #include <sys/stat.h>
3 #include <stdlib.h>
4 #include <time.h>
5 int main(int argc, char*argv[]){
6     struct stat
7     file; int n;
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <filename>\n"); exit(-1);
11     }
12     if ((n = stat(argv[1], &file)) == -1)
13     {
14         perror(argv[1]);
15         exit(-1);
16     }
17     printf("User id : %d\n", file.st_uid);
18     printf("Group id : %d\n", file.st_gid);
19     printf("Block size : %d\n", file.st_blksize);
20     printf("Blocks allocated : %d\n", file.st_blocks);
21     printf("Inode no. : %d\n", file.st_ino);
22     printf("Last accessed : %s", ctime(&(file.st_atime)));
23     printf("Last modified : %s", ctime(&(file.st_mtime)));
24     printf("File size : %d bytes\n", file.st_nlink);
25     printf("Permissions :");
26     printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
27     printf( (file.st_mode & S_IRUSR) ? "r" : "-");
28     printf( (file.st_mode & S_IWUSR) ? "w" : "-");
29     printf( (file.st_mode & S_IXUSR) ? "x" : "-");
30     printf( (file.st_mode & S_IRGRP) ? "r" : "-");
31     printf( (file.st_mode & S_IWGRP) ? "w" : "-");
32     printf( (file.st_mode & S_IXGRP) ? "x" : "-");
33     printf( (file.st_mode & S_IROTH) ? "r" : "-");
34     printf( (file.st_mode & S_IWOTH) ? "w" : "-");
35     printf( (file.st_mode & S_IXOTH) ? "x" : "-");
```

Output



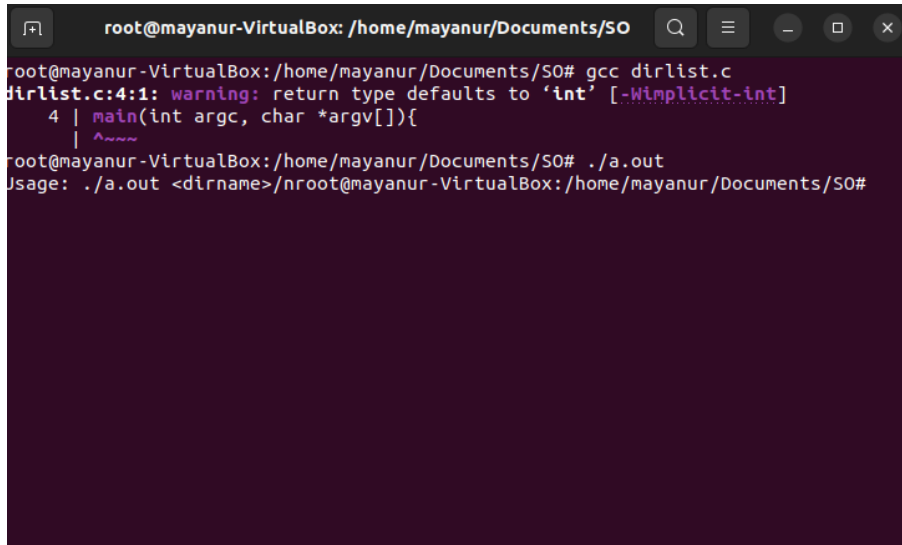
```
root@mayanur-VirtualBox: /home/mayanur/Documents/SO# gcc stat.c
stat.c: In function 'main':
stat.c:7:9: error: storage size of 'file' isn't known
    7 |         file; int n;
      |         ^~~~~
root@mayanur-VirtualBox: /home/mayanur/Documents/SO# ./a.out
Usage: ./a.out <S0>
root@mayanur-VirtualBox: /home/mayanur/Documents/SO#
```

5. DIRLIST.C



```
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <stdlib.h>
4 main(int argc, char *argv[]){
5     struct dirent *dptr;
6     DIR *dname;
7
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <dirname>\n");
11         exit(-1);
12     }
13     if((dname = opendir(argv[1])) == NULL)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     while(dptr=readdir(dname))
19         printf("%s\n", dptr->d_name);
20
21     closedir(dname);
22 }
```

Output



```
root@mayanur-VirtualBox: /home/mayanur/Documents/SO
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
4 | main(int argc, char *argv[]){
  | ^~~~~
root@mayanur-VirtualBox:/home/mayanur/Documents/SO# ./a.out
Usage: ./a.out <dirname>/nroot@mayanur-VirtualBox:/home/mayanur/Documents/SO#
```