```
#include<stdio.h>
#include<sys/types.h>
structures
#include<unistd.h>
#include<sys/wait.h>

void main()
{       int status,pid,childpid;
        pid=fork();
        if (pid==-1)
        {       printf("Child Process creation failed");
                return;         }
        else if (pid==0)
        {       printf("Child Process created with ID: %d\n",getpid());
                char *arg[]={"Hello",NULL);
                execvp("./child", arg);       }
        else
        {       childpid = wait(&status);
                printf("Parent Process created with ID: %d\n", getpid());
                printf("Child Process created successfully");  }}
```

```
#include<stdio.h>
#include<sys/stat.h>
#include<time.h>
#include<stdlib.h>
void main()
{       char[50];
        struct stat *node;
        node=(struct stat*)malloc(sizeof(struct stat));
        printf("Enter the filename: ");
        scanf("%s",file);
        stat(file,node);
        if(node->st_ino==0)
        {       printf("FILE DOES NOT EXIST");  }
        else
        {       printf("\nInode/Serial number: %d",node->st_ino);
                printf("\nBlock Size: %d",node->st_blksize);
                printf("\nAccess Time: %d,node->st_atime);
                printf("\nLast Modified Time: %d,node->st_mtime);
                printf("\nGroup Id: %d",node->st_gid);
                printf("\nSize of File: %d",node->st_size);
                printf("\nPermissions: %d",node->st_mode);
                printf("\nUserId: %d",node->st_uid);      }}
```

```
#include<stdio.h>

#include<sys/stat.h>

#include<dirent.h>


void main()

{       DIR *dir; struct dirent *ptr2; char dir_name[50];

        printf("Enter Directory: "); scanf("%s",dir_name);

        dir=opendir(dir_name);

        while((ptr2=readdir(dir))!=NULL)

        {       printf("%ID\t%s\n",ptr->d_ino,ptr2->d_name);    }

        closedir(dirr);

}
```