File Locking:

1. **File Description:** First, you are supposed to generate a fixed length, fixed format file. There should be exactly 72 lines in the file. Each line containing two columns. First column should contain five bytes of data: "PID " Second column should contain five bytes data: BLNK followed by new line character. First column and second column should be separated by a blank character. This format mimics 72 berths in a railway coach. In any given line, BNLK denotes that particular berth is available for reservation. The file should contain exactly (5 + 1 + 5) * 72 = 792 bytes of data.

   Of these 72 berths, every consecutive eight berths are reserved for traveling to a particular railway station say:

Station	Berth Numbers
KZJ (Kazipet)	1 - 8
RDM (Ramagundam)	9 - 16
BPQ (Ballarshah)	17 - 24
NGP (Nagpur)	25 - 32
BPL (Bhopal)	33 - 40
JHS (Jhansi)	41 - 48
GWL (Gwalior)	49 - 56
AGC (Agra)	57 - 64
NDLS (New Delhi)	65 - 72

2. **Your task** is:
   - To create one child process per destination station (using `fork`).
   - Lock *only* those berths which are going to destination station (that is only those 8 * (5 + 1 + 5) = 88 bytes where this child process is traveling).
   - Find out whether berths are available for this process to travel or not (BLNK is present or not for those eight berths).
   - If a berth is available, reserve a berth to this child process. Which means, you should write the process ID of the child (using: `getpid()`) followed by a blank followed by destination station code.

     NOTE: While performing writing operation, total number of bytes in the file should not change.
   - Once reservation is completed, unlock the bytes the child process has locked.
   - Exit child process.
3. **Input:** Fixed length, fixed format file that you have created through the program.
4. **Example Output**: The **above** fixed length, fixed format file that you have created should read as follows

19005	blank	NGP
19007	blank	NGP
19008	blank	NGP
19009	blank	NGP
PID	blank	BLNK
PID	blank	BLNK
PID	blank	BLNK
PID	blank	BLNK

5. **Notes:**
   (a) You should implement this program either in **C** or **C++** language only.
   (b) To get process ID of a child use the system call: `getpid()`.
   (c) Material shared with you will help you implement this problem. In particular chapter 55.
   (d) TAs will starting taking *Attendance* from 2:30 PM on wards.
   (e) Lab duration 2:00 PM to 5:30 PM. Evaluation starts from 5:30 PM on wards.
6. **Marks Distribution:**

Evaluation Point	Description	Marks
1	Creation of input data file	1
2	(Un) Locking exact bytes of given file	2
3	Writing data with specified format	2
4	Child process work	2
5	Logic	3