

Struct and function:

Task1 & Task2:

```
~/a/c/a/struct_and_function
→ assignment2 ls
struct_and_function/  system_Call/  threading/
→ assignment2 cd struct_and_function/
→ struct_and_function ls
task1* task1.c task2* task2.c
→ struct_and_function gcc ./task1.c -o task1
→ struct_and_function ./task1
Quantity of Paratha: 25
Unit Price: 10
Quantity of Vegetable: 5
Unit Price: 20
Quantity of Water: 20
Unit Price: 20
Number of people: 6
Individual people will pay: 125.00 tk
→ struct_and_function gcc ./task2.c -o task2
→ struct_and_function ./task2
lower bound: 6
upper bound: 28
6
28
→ struct_and_function ./task2
lower bound: 1
upper bound: 10000
6
28
496
8128
→ struct_and_function []
```

System Call:

Task1, Task2, Task3

```
~/a/c/a/system_Call
→ struct_and_function
→ struct_and_function cd ..
→ assignment2 cd system_Call/
→ system_Call ls
inputfile1.txt  oddeven.c  sort.c  task1.c  task2.c  task3.c  task4.c  task5.c
oddeven*      sort*      task1*  task2*  task3*  task4*  task5*
→ system_Call gcc ./task1.c -o task1
→ system_Call ./task1
Input: ^C
→ system_Call ./task1 inputfile2.txt
Input: Hi
Input: this is input file 2
Input: from the task1
Input: -1
→ system_Call cat inputfile2.txt
Hi
this is input file 2
from the task1
→ system_Call gcc ./task2.c -o task2
→ system_Call ./task2
I am grandchild
I am child
I am parent
→ system_Call gcc ./task3.c -o task3
→ system_Call ./task3
Total processes created: 6
→ system_Call
```

Task4 and Task5:

```
~/a/c/a/system_Call
→ system_Call gcc ./task4.c -o task4
→ system_Call ./task4 12 23 42 22 33 122 456 432
456 432 122 42 33 23 22 12
12 : even
23 : odd
42 : even
22 : even
33 : odd
122 : even
456 : even
432 : even
→ system_Call gcc ./task5.c -o task5
→ system_Call ./task5
Parent process ID: 67192
Child process ID: 67193
Grandchild process ID: 67194
Grandchild process ID: 67195
Grandchild process ID: 67196
→ system_Call █
```

Threading:

Task1 and Task2 and Task3:

```
~/a/c/a/threading
Grandchild process ID: 67196
→ system_Call cd ../threading/
→ threading gcc ./task1.c -o task1
→ threading ./task1
thread-1 running
thread-1 closed
thread-2 running
thread-2 closed
thread-3 running
thread-3 closed
thread-4 running
thread-4 closed
thread-5 running
thread-5 closed
→ threading gcc ./task2.c -o task2
→ threading ./task2
thread-1 prints 1
thread-1 prints 2
thread-1 prints 3
thread-1 prints 4
thread-1 prints 5
thread-2 prints 6
thread-2 prints 7
thread-2 prints 8
thread-2 prints 9
thread-2 prints 10
thread-3 prints 11
thread-3 prints 12
thread-3 prints 13
thread-3 prints 14
thread-3 prints 15
thread-4 prints 16
thread-4 prints 17
thread-4 prints 18
thread-4 prints 19
thread-4 prints 20
thread-5 prints 21
thread-5 prints 22
thread-5 prints 23
thread-5 prints 24
thread-5 prints 25
→ threading gcc ./task3.c -o task3
→ threading ./task3
asd
ads
af
Miracle
→ threading █
```