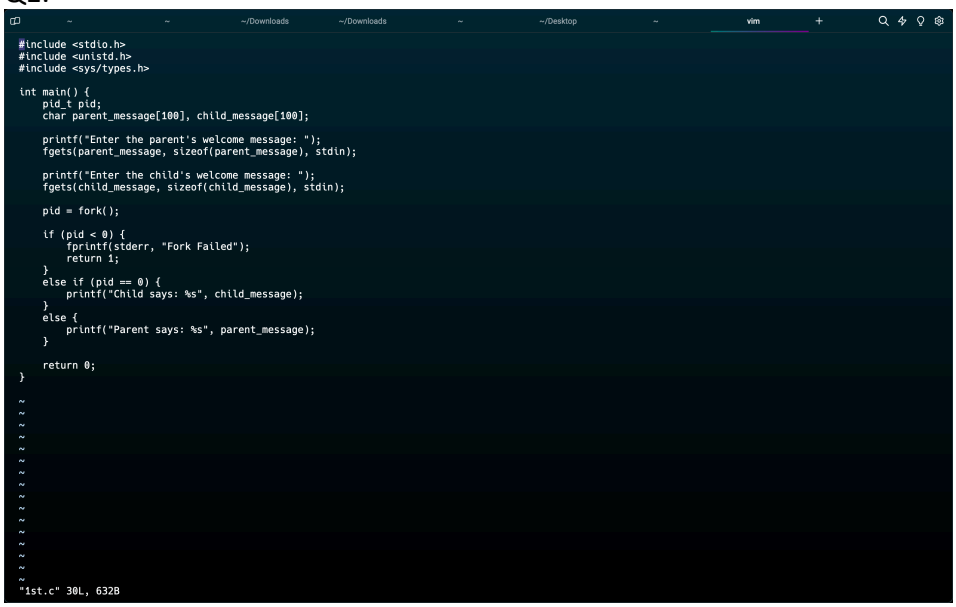
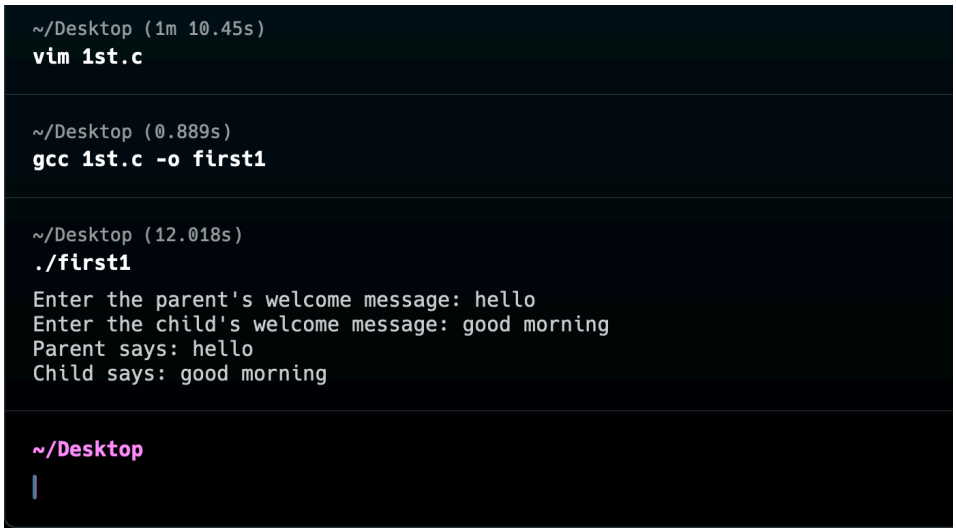
VIKRAM MALHOTRA
E22CSEU0735
BATCH-26

Q1.

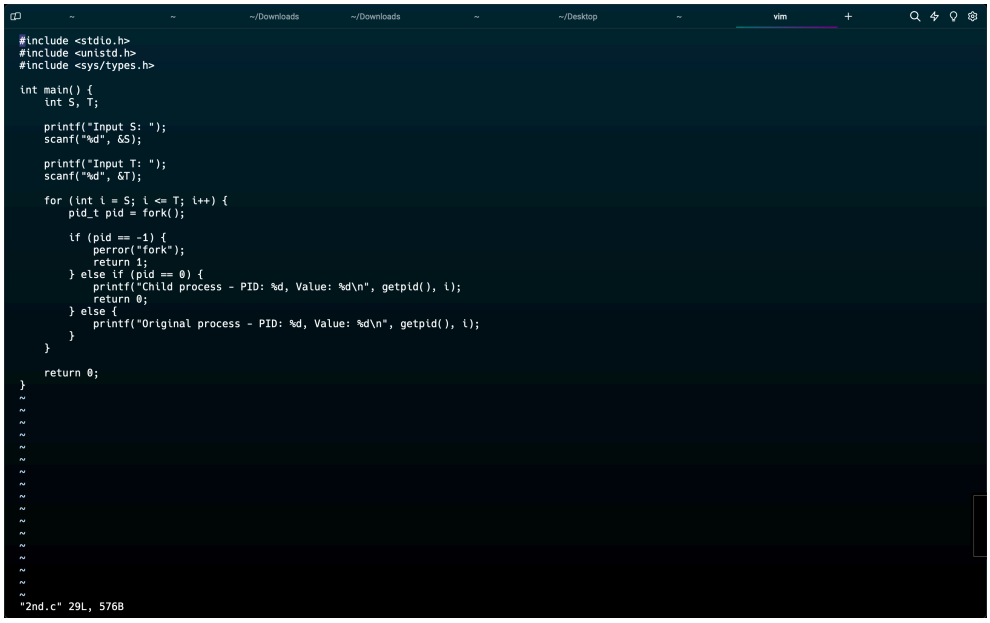


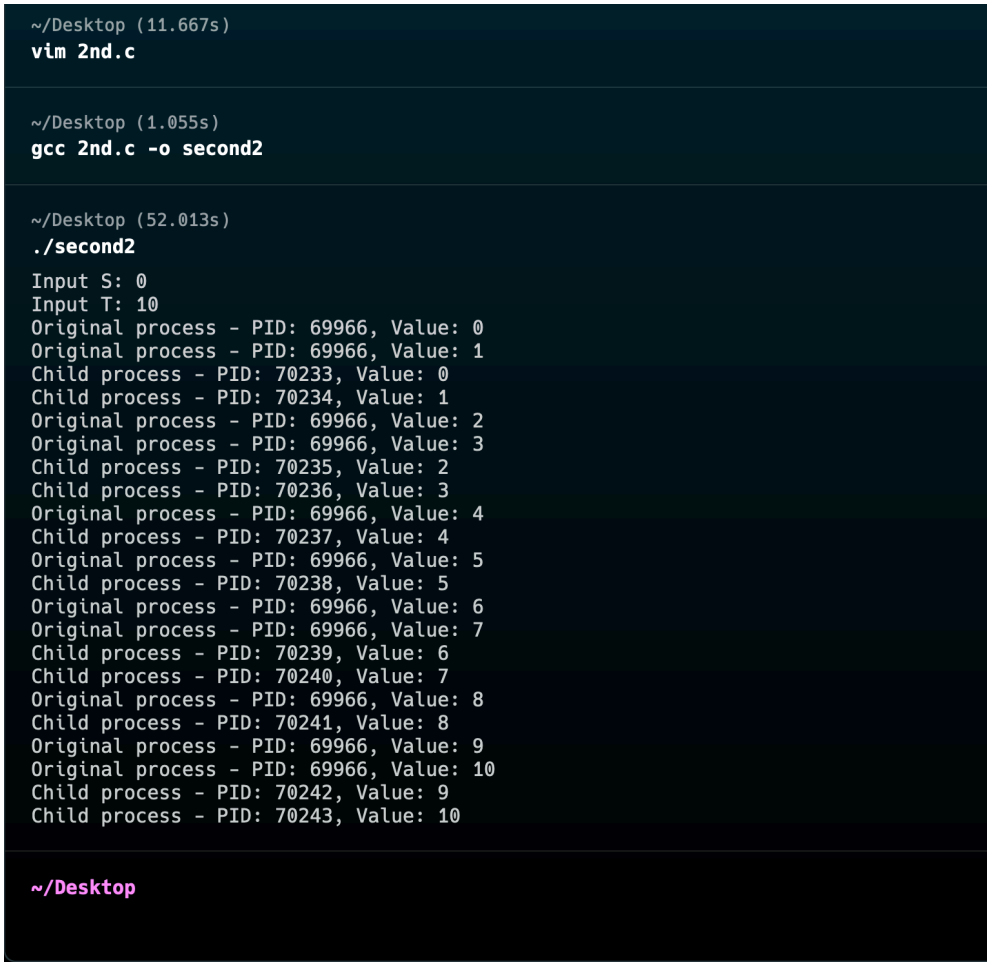
```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    pid_t pid;
    char parent_message[100], child_message[100];

    printf("Enter the parent's welcome message: ");
    fgets(parent_message, sizeof(parent_message), stdin);

    printf("Enter the child's welcome message: ");
    fgets(child_message, sizeof(child_message), stdin);

    pid = fork();

    if (pid < 0) {
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) {
        printf("Child says: %s", child_message);
    }
    else {
        printf("Parent says: %s", parent_message);
    }

    return 0;
}
```



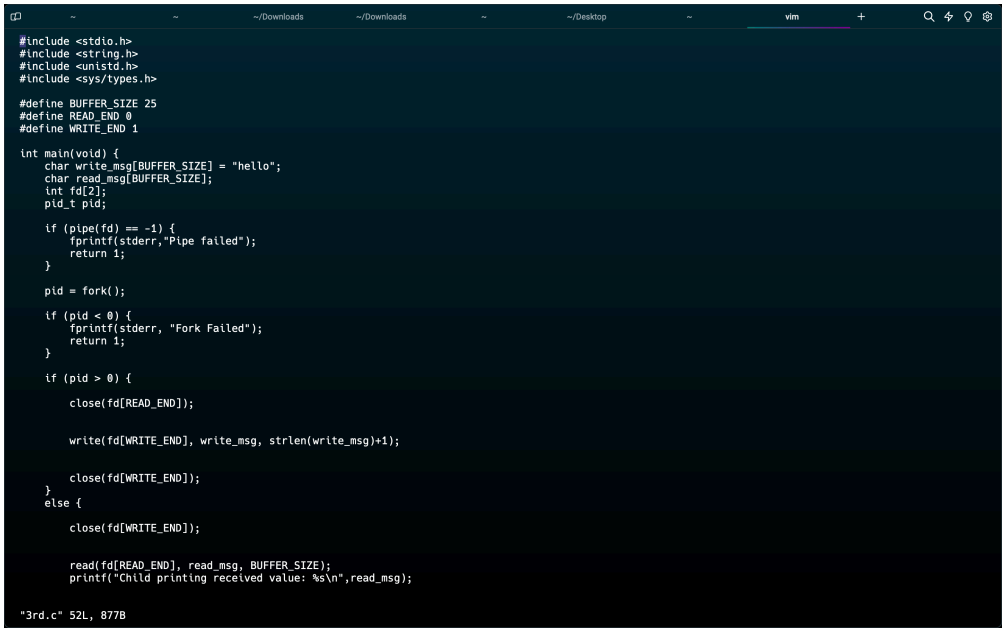
```
~/Desktop (1m 10.45s)
vim 1st.c

~/Desktop (0.889s)
gcc 1st.c -o first1

~/Desktop (12.018s)
./first1
Enter the parent's welcome message: hello
Enter the child's welcome message: good morning
Parent says: hello
Child says: good morning

~/Desktop
```

Q2.


~/Downloads
~/Downloads
~/Desktop
vim
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
int S, T;
printf("Input S: ");
scanf("%d", &S);
printf("Input T: ");
scanf("%d", &T);
for (int i = S; i <= T; i++) {
pid_t pid = fork();
if (pid == -1) {
perror("fork");
return 1;
} else if (pid == 0) {
printf("Child process - PID: %d, Value: %d\n", getpid(), i);
return 0;
} else {
printf("Original process - PID: %d, Value: %d\n", getpid(), i);
}
}
return 0;
}
"2nd.c" 29L, 576B


~/Desktop (11.667s)
vim 2nd.c
~/Desktop (1.055s)
gcc 2nd.c -o second2
~/Desktop (52.013s)
./second2
Input S: 0
Input T: 10
Original process - PID: 69966, Value: 0
Original process - PID: 69966, Value: 1
Child process - PID: 70233, Value: 0
Child process - PID: 70234, Value: 1
Original process - PID: 69966, Value: 2
Original process - PID: 69966, Value: 3
Child process - PID: 70235, Value: 2
Child process - PID: 70236, Value: 3
Original process - PID: 69966, Value: 4
Child process - PID: 70237, Value: 4
Original process - PID: 69966, Value: 5
Child process - PID: 70238, Value: 5
Original process - PID: 69966, Value: 6
Original process - PID: 69966, Value: 7
Child process - PID: 70239, Value: 6
Child process - PID: 70240, Value: 7
Original process - PID: 69966, Value: 8
Child process - PID: 70241, Value: 8
Original process - PID: 69966, Value: 9
Original process - PID: 69966, Value: 10
Child process - PID: 70242, Value: 9
Child process - PID: 70243, Value: 10
~/Desktop

Q3.


~
~
~/Downloads
~/Downloads
~
~/Desktop
~
vim
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1
int main(void) {
char write_msg[BUFFER_SIZE] = "hello";
char read_msg[BUFFER_SIZE];
int fd[2];
pid_t pid;
if (pipe(fd) == -1) {
fprintf(stderr,"Pipe failed");
return 1;
}
pid = fork();
if (pid < 0) {
fprintf(stderr, "Fork Failed");
return 1;
}
if (pid > 0) {
close(fd[READ_END]);
write(fd[WRITE_END], write_msg, strlen(write_msg)+1);
close(fd[WRITE_END]);
}
else {
close(fd[WRITE_END]);
read(fd[READ_END], read_msg, BUFFER_SIZE);
printf("Child printing received value: %s\n",read_msg);
"3rd.c" 52L, 877B


~/Desktop (3m 38.71s)
vim 3rd.c
~/Desktop (1.073s)
gcc 3rd.c -o third3
~/Desktop (0.593s)
./third3
Child printing received value: hello
~/Desktop