

question 1aa



```
#include <sys/types.h>
#include <sys/wait.h>
int main(int argc, char *argv[])
{ // (d) Check for correct number of arguments if
  (argc != 3) { fprintf(stderr, "Usage: %s \n", argv[0]);
    exit(1); } // (i) Print from the parent process
  printf("Parent Process ID: %d\n", getpid());
  printf("Parent's Parent Process ID: %d\n", getppid());
  printf("Parent Process Group ID: %d\n", getpgrp());
  pid_t pid = fork(); // (e) Fork if (pid < 0) {
    perror("fork"); exit(1); } else if (pid == 0) { // This
    block will be executed by child // (h) Print from the
    child process printf("\nChild Process ID: %d\n",
      getpid()); printf("Child's Parent Process ID: %d\n",
      getppid()); printf("Child Process Group ID: %d\n",
      getpgrp()); // (f) Use execl to call cp // Uncomment
    below line for execl // execl("/bin/cp", "cp", "-p", "-i",
    argv[1], argv[2], (char *)NULL); // (j) Comment out
    the execl call and add instead a call to execv char
    *newargv[] = {"cp", "-p", "-i", argv[1], argv[2], NULL};
    execv("/bin/cp", newargv); // execv call
    perror("exec"); // execv or execl failed exit(1); } else {
    // This block will be executed by parent wait(NULL);
    // (g) Wait for the child to finish } return 0; }
```

[Read More](#)