🔬

OS LAB 6

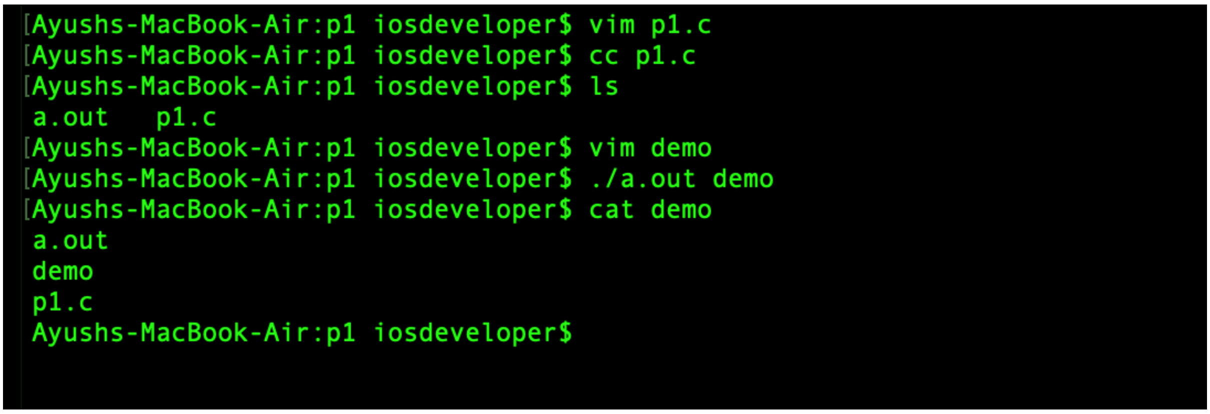
`Ayush Singh`

`209301163`

1. Write a program to demonstrate redirection.

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[])
{
char d[50];
if(argc==2)
{
bzero(d,sizeof(d));
strcat(d,"ls ");
strcat(d,"> ");
strcat(d,argv[1]);
system(d);
}
else
printf("\nInvalid No. of inputs");
}
```

Execution

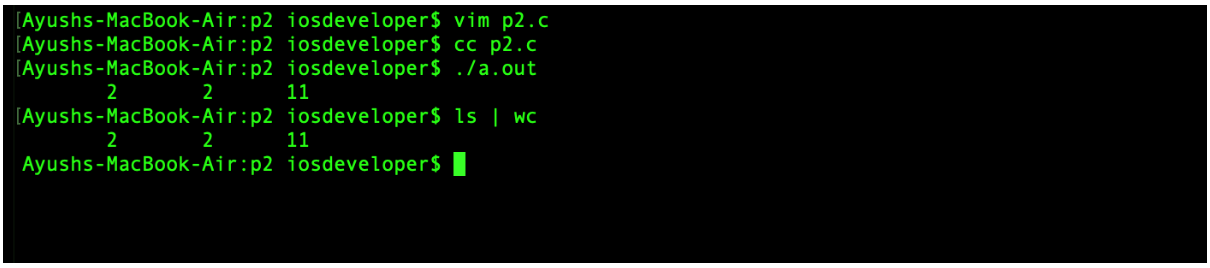


```
[Ayushs-MacBook-Air:p1 iosdeveloper$ vim p1.c
[Ayushs-MacBook-Air:p1 iosdeveloper$ cc p1.c
[Ayushs-MacBook-Air:p1 iosdeveloper$ ls
 a.out    p1.c
[Ayushs-MacBook-Air:p1 iosdeveloper$ vim demo
[Ayushs-MacBook-Air:p1 iosdeveloper$ ./a.out demo
[Ayushs-MacBook-Air:p1 iosdeveloper$ cat demo
 a.out
 demo
 p1.c
 Ayushs-MacBook-Air:p1 iosdeveloper$
```

2. Write a program to implement ls | wc using pipes.

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include<errno.h>
#include<sys/wait.h>
#include <unistd.h>
int main(){

  int a[2];

  pipe(a);

  if(!fork())
  {
    close(1);

    dup(a[1]);

    close(a[0]);

    execlp("ls","ls",NULL);
  }
  else
  {
    close(0);

    dup(a[0]);

    close(a[1]);

    execlp("wc","wc",NULL);
  }
}
```

Execution

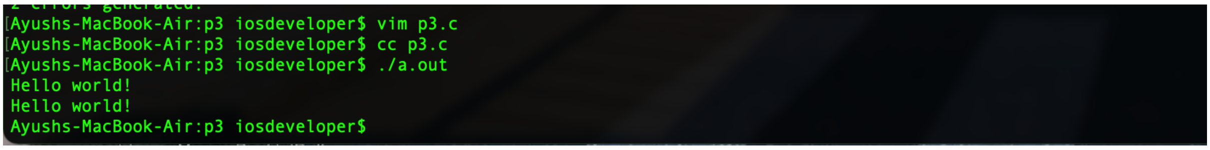


```
[Ayushs-MacBook-Air:p2 iosdeveloper$ vim p2.c
[Ayushs-MacBook-Air:p2 iosdeveloper$ cc p2.c
[Ayushs-MacBook-Air:p2 iosdeveloper$ ./a.out
       2       2      11
[Ayushs-MacBook-Air:p2 iosdeveloper$ ls | wc
       2       2      11
 Ayushs-MacBook-Air:p2 iosdeveloper$
```

3. Write a program to demonstrate pipe for the scenario where child process sends a message hello to parent process parent process receives message and display on screen.

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
 int main()
{
    fork();
    printf("Hello world!\n");
    return 0;
}
```

Execution

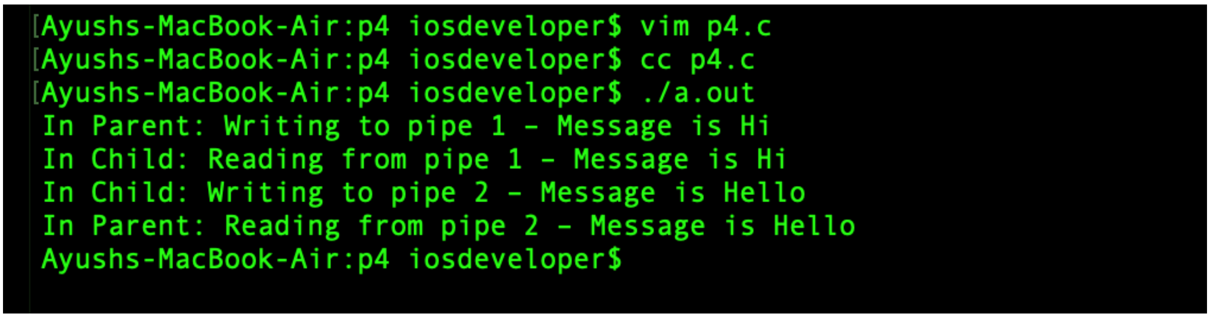


```
2 errors generated.
[Ayushs-MacBook-Air:p3 iosdeveloper$ vim p3.c
[Ayushs-MacBook-Air:p3 iosdeveloper$ cc p3.c
[Ayushs-MacBook-Air:p3 iosdeveloper$ ./a.out
Hello world!
Hello world!
Ayushs-MacBook-Air:p3 iosdeveloper$
```

4. Write a program to demonstrate the use of double pipes

```
#include<stdio.h>
#include<unistd.h>

int main() {
   int pipefds1[2], pipefds2[2];
   int returnstatus1, returnstatus2;
   int pid;
   char pipe1writemessage[20] = "Hi";
   char pipe2writemessage[20] = "Hello";
   char readmessage[20];
   returnstatus1 = pipe(pipefds1);

   if (returnstatus1 == -1) {
      printf("Unable to create pipe 1 \n");
      return 1;
   }
   returnstatus2 = pipe(pipefds2);
   if (returnstatus2 == -1) {
      printf("Unable to create pipe 2 \n");
      return 1;
   }
   pid = fork();

   if (pid != 0) {
      close(pipefds1[0]);
      close(pipefds2[1]);
      printf("In Parent: Writing to pipe 1 – Message is %s\n", pipe1writemessage);
      write(pipefds1[1], pipe1writemessage, sizeof(pipe1writemessage));
      read(pipefds2[0], readmessage, sizeof(readmessage));
      printf("In Parent: Reading from pipe 2 – Message is %s\n", readmessage);
   } else {
      close(pipefds1[1]);
      close(pipefds2[0]);
      read(pipefds1[0], readmessage, sizeof(readmessage));
      printf("In Child: Reading from pipe 1 – Message is %s\n", readmessage);
      printf("In Child: Writing to pipe 2 – Message is %s\n", pipe2writemessage);
      write(pipefds2[1], pipe2writemessage, sizeof(pipe2writemessage));
   }
   return 0;
}
```

Execution



```
[Ayushs-MacBook-Air:p4 iosdeveloper$ vim p4.c
[Ayushs-MacBook-Air:p4 iosdeveloper$ cc p4.c
[Ayushs-MacBook-Air:p4 iosdeveloper$ ./a.out
 In Parent: Writing to pipe 1 - Message is Hi
 In Child: Reading from pipe 1 - Message is Hi
 In Child: Writing to pipe 2 - Message is Hello
 In Parent: Reading from pipe 2 - Message is Hello
 Ayushs-MacBook-Air:p4 iosdeveloper$
```