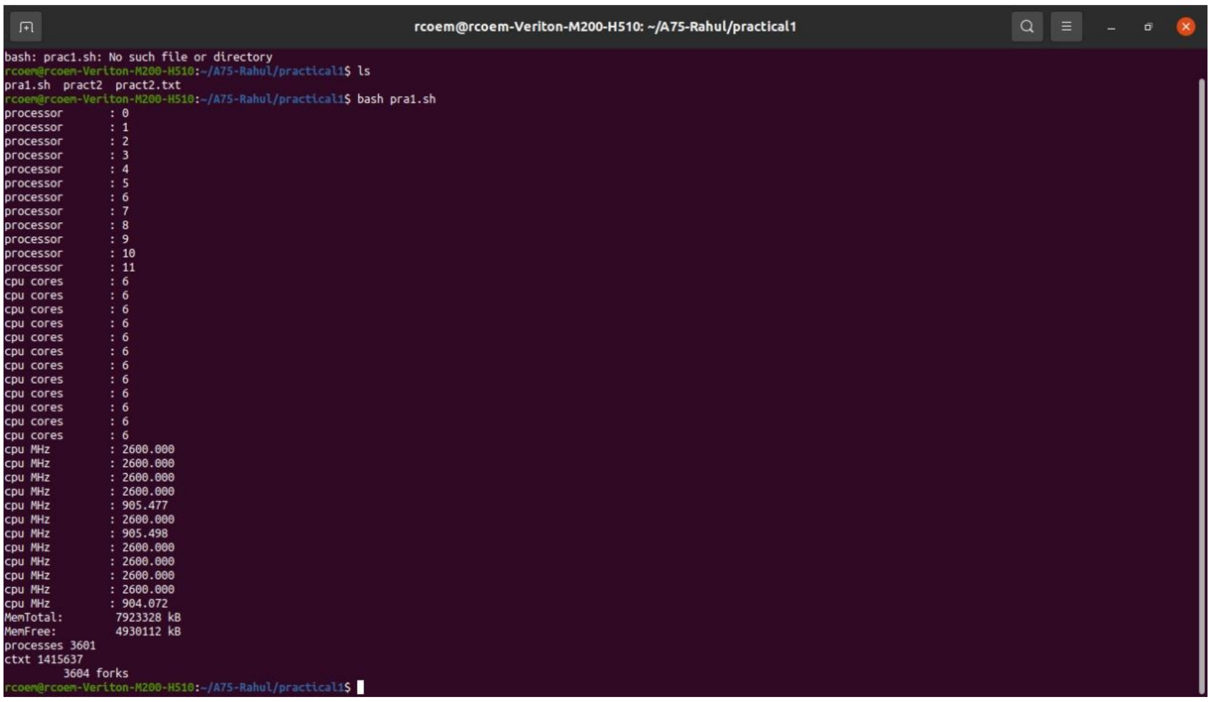
Shell Script

```
cd /
cd proc
cat cpuinfo | grep -h processor
cat cpuinfo | grep -h cores
cat cpuinfo | grep -h 'cpu MHz'
cat meminfo | grep -h MemTotal
cat meminfo | grep -h MemFree
cat stat | grep -h processes
cat stat | grep -h ctxt
vmstat –f
```

Output



```
bash: prac1.sh: No such file or directory
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical1$ ls
pra1.sh  pract2  pract2.txt
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical1$ bash pra1.sh
processor       : 0
processor       : 1
processor       : 2
processor       : 3
processor       : 4
processor       : 5
processor       : 6
processor       : 7
processor       : 8
processor       : 9
processor       : 10
processor       : 11
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu cores       : 6
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 905.477
cpu MHz         : 2600.000
cpu MHz         : 905.498
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 2600.000
cpu MHz         : 904.072
MemTotal:        7923328 kB
MemFree:         4930112 kB
processes 3601
ctxt 1415637
         3604 forks
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical1$ 
```

Code

Program

```
#include<stdio.h>
#include<fcntl.h>
#include<stdlib.h>
#include<errno.h>
#include<unistd.h>
#include<string.h>
#include<sys/types.h>
#include<sys/stat.h>

int main(){
     int c, fd;
     size_t sz;
     char *c1 = (char *)calloc(100, sizeof(char));
     char con[100];
    struct stat buf;
    char *p;
    char pat[100];
     int status;

     while(c != 10){
         printf("1. Create file\n");
         printf("2. Open file\n");
         printf("3. Write file and size of file\n");
         printf("4. Read file and size of file\n");
         printf("5. Print file content in reverse\n");
         printf("6. Delete file\n");
         printf("7. Status(stat)\n");
         printf("8. Status(fstat)\n");
         printf("9. Search pattern\n");
         printf("10. Exit\n");
         printf("Enter your choice :");
         scanf("%d", &c);

         switch(c){
          case 1:
               fd = creat("f1.txt",0770);
               printf("File is created\n");

               printf("fd = %d\n", fd);
```

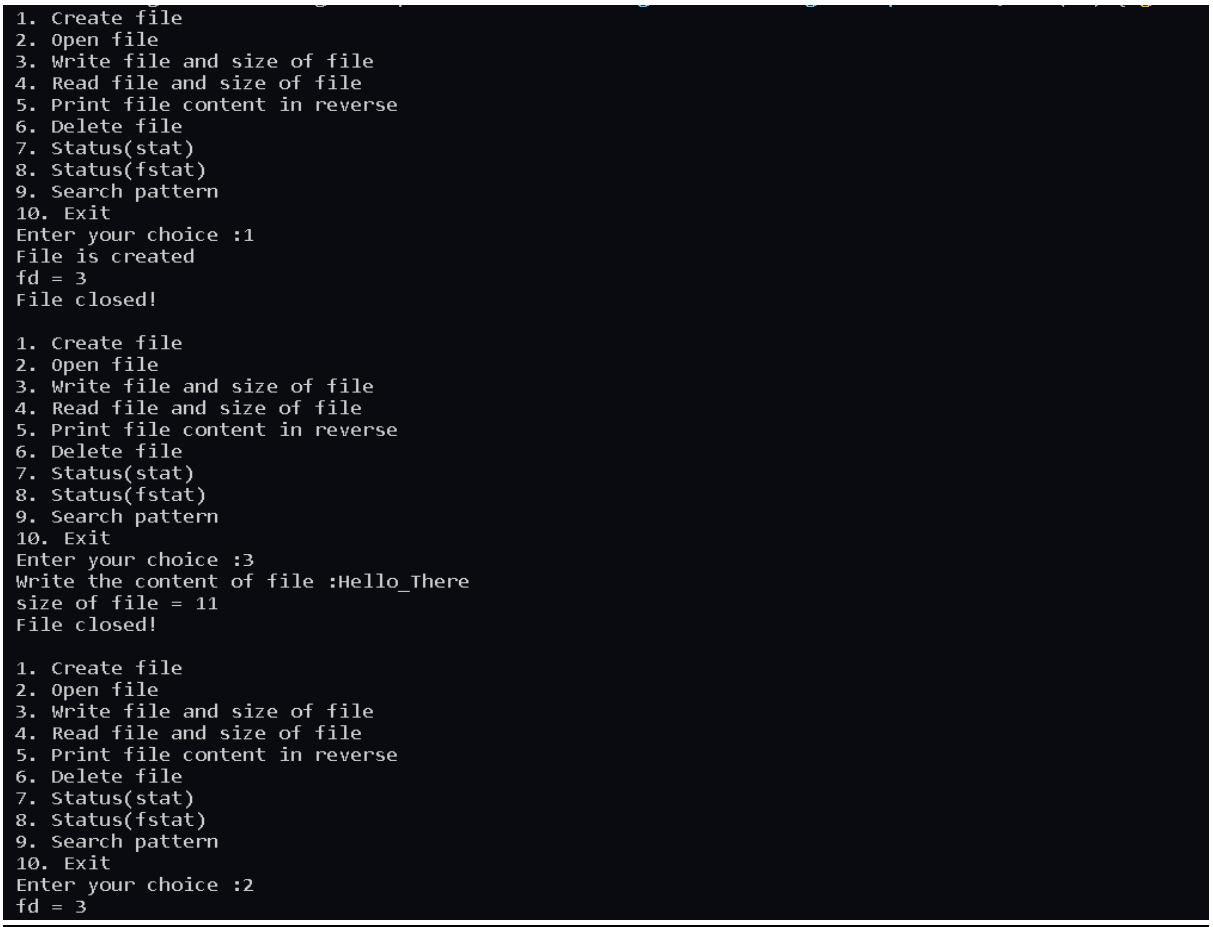
```
            close(fd);
            printf("File closed!\n\n");
            break;

        case 2:
            fd = open("f1.txt", O_RDONLY|O_CREAT, 0777);

            printf("fd = %d\n", fd);

            close(fd);
            printf("File closed!\n\n");
            break;

        case 3:
            fd = open("f1.txt", O_WRONLY|O_CREAT, 0777);
            printf("Write the content of file :");
            scanf("%s", con);

            sz = write(fd, con, strlen(con));
            printf("size of file = %ld\n", sz);

            close(fd);
            printf("File closed!\n\n");
            break;

        case 4:
            fd = open("f1.txt", O_RDONLY);

            sz = read(fd, c1, 100);
            printf("size of file is :%ld\n", sz);
            printf("fd = %d\n", fd);

            c1[sz] = '\0';
            printf("%s\n", c1);

            close(fd);
            printf("File closed!\n\n");
            break;

        case 5:
            fd = open("f1.txt", O_RDONLY);
```

```
                sz = read(fd, c1, 100);
                printf("fd = %d\n", fd);

                c1[sz] = '\0';
                int i = 100;
                while(i != -1){
                     printf("%c", c1[i]);
                     i--;
                }
                printf("\n");

                close(fd);
                printf("File closed!\n\n");
                break;

           case 6:
                unlink("f1.txt");
                printf("File deleted!\n\n");
                break;

           case 7:
                stat("f1.txt", &buf);
                printf("file mode :%d\nfile serial number :%ld\nfile
size :%d\nuser ID :%d\ngroup ID :%d\n", buf.st_mode, buf.st_ino,
buf.st_size, buf.st_uid, buf.st_gid);
                break;

            case 8:
                status = fstat(fd, &buf);
                printf("Status :%d\n", status);
                printf("file mode :%d\nfile serial number :%ld\nfile
size :%d\nuser ID :%d\ngroup ID :%d\n", buf.st_mode, buf.st_ino,
buf.st_size, buf.st_uid, buf.st_gid);
                break;

            case 9:
                fd = open("f1.txt",O_RDONLY, 0777);
                sz = lseek(fd, 0, SEEK_END);
                lseek(fd, 0, SEEK_SET);
                printf("size of file is :%ld\n", sz);
                read(fd, c1, sz);
                c1[sz] = '\0';
                printf("Enter pattern :");
```

```
                    scanf("%s", pat);
                    p = strstr(c1, pat);
                    if(p)
                        printf("\nString Found\n");
                    else
                        printf("\nString Not Found\n");
                    break;

                case 10:
                    break;

               default:
                    printf("Invalid choice!\n");
             }
        }
}
```

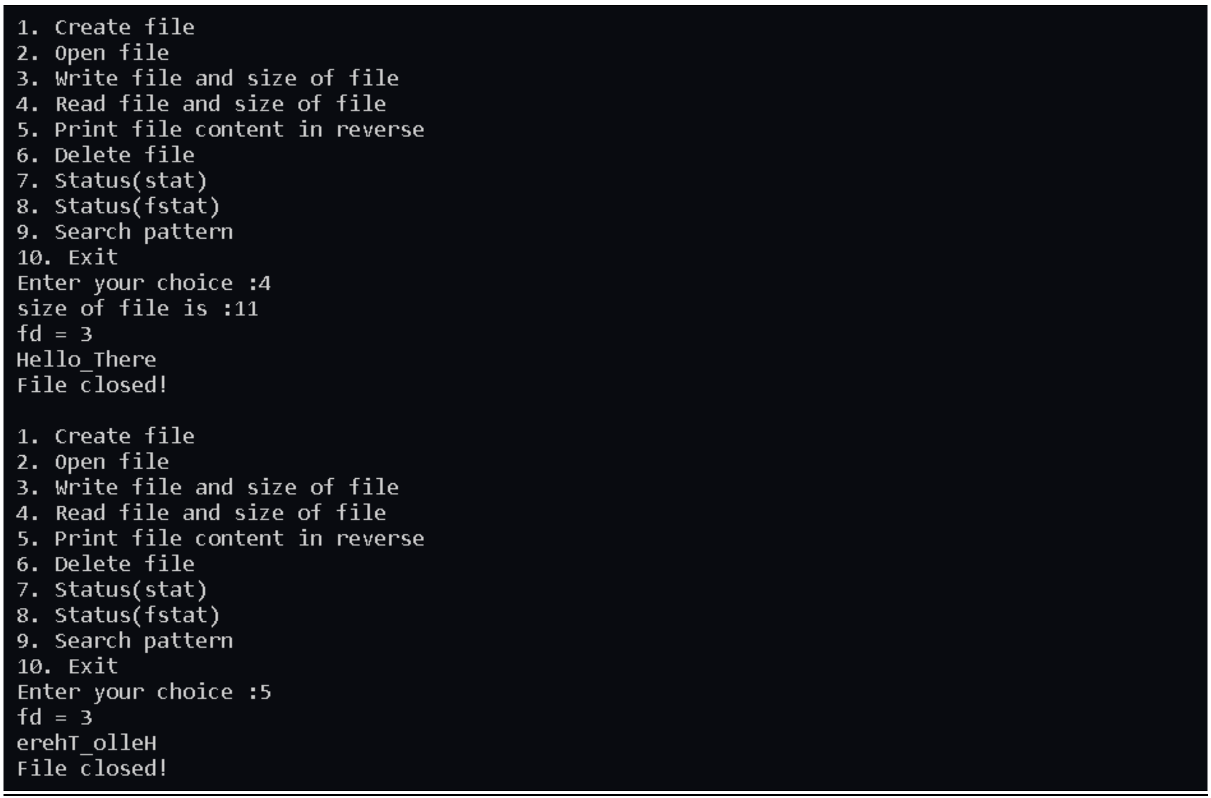
<u>Output</u>

Creating a file and writing a string in file



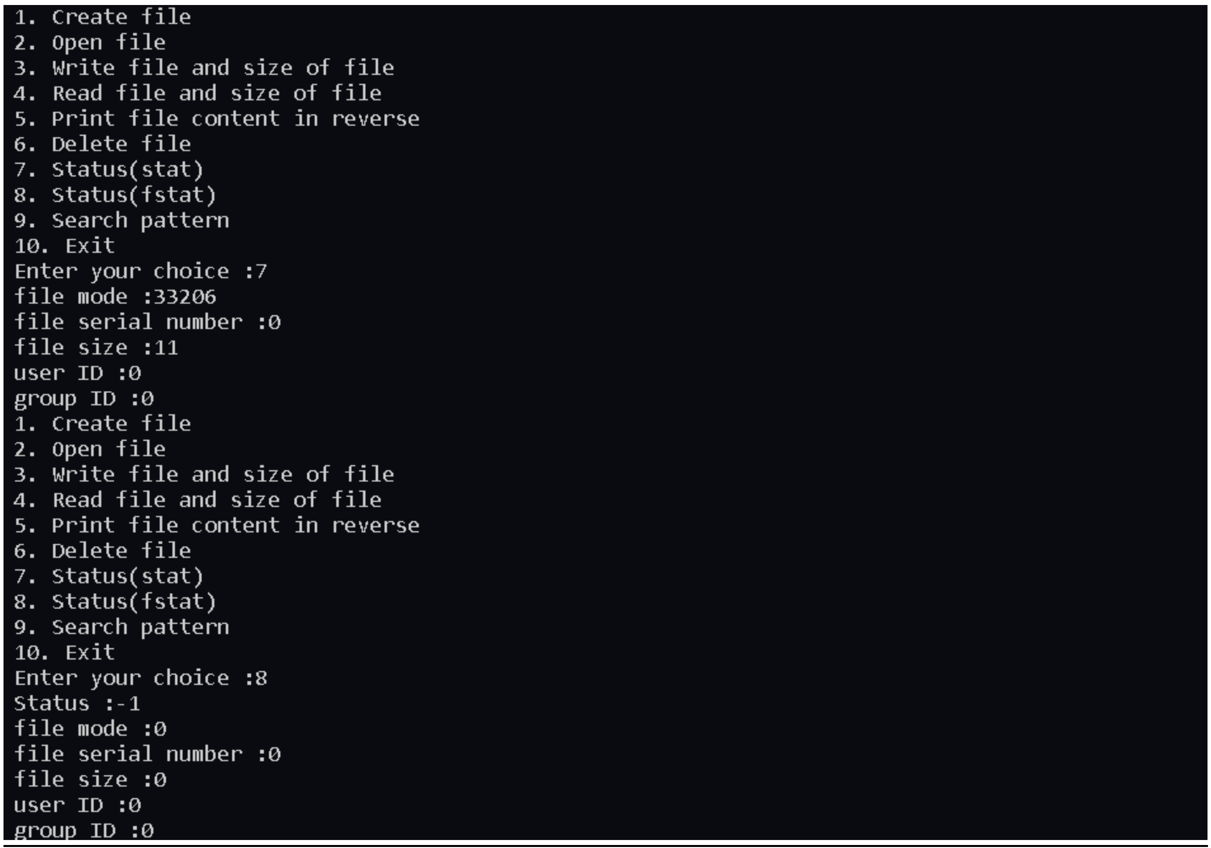
```
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :1
File is created
fd = 3
File closed!

1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :3
Write the content of file :Hello_There
size of file = 11
File closed!

1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :2
fd = 3
```

Reading a file and print the content in reverse order



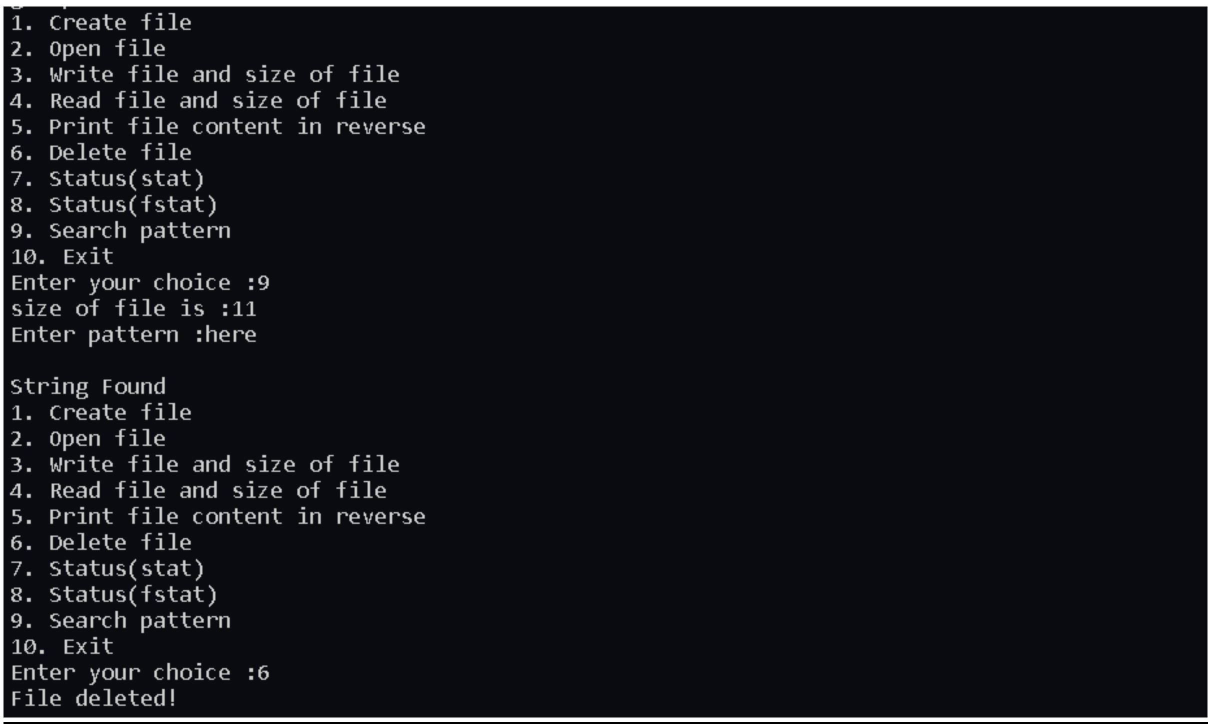
```
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :4
size of file is :11
fd = 3
Hello_There
File closed!

1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :5
fd = 3
erehT_olleH
File closed!
```

Printing Status of a file using stat and fstat



```
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :7
file mode :33206
file serial number :0
file size :11
user ID :0
group ID :0
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :8
Status :-1
file mode :0
file serial number :0
file size :0
user ID :0
group ID :0
```

Searching pattern in file and deleting a file

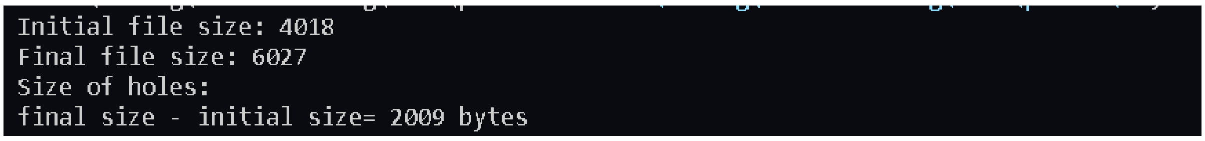


```
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :9
size of file is :11
Enter pattern :here

String Found
1. Create file
2. Open file
3. Write file and size of file
4. Read file and size of file
5. Print file content in reverse
6. Delete file
7. Status(stat)
8. Status(fstat)
9. Search pattern
10. Exit
Enter your choice :6
File deleted!
```

Assignment 1

```
#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<errno.h>
#include<unistd.h>
#include<string.h>
#include<sys/types.h>
#include<sys/stat.h>
int main(){
    int fd;
    char str[2000];
    int size;
    int fsize;
    struct stat st;
    size_t sz;

    fd=open("file1.txt",O_WRONLY|O_CREAT,0777);
    stat("file1.txt", &st);
    size = st.st_size;
    printf("Initial file size: %d\n", size);
    sz = write(fd, str, 2000);
    lseek(fd, 0, SEEK_END);
    sz = write(fd, str, 2000);
    close(fd);
    fd=open("file1.txt",O_RDONLY,0777);
    stat("file1.txt", &st);
    fsize = st.st_size;
    printf("Final file size: %d\n", fsize);
    printf("Size of holes:\nfinal size - initial size= %d bytes\n",
fsize-size);
    close(fd);
    return 0;
}
```

Output

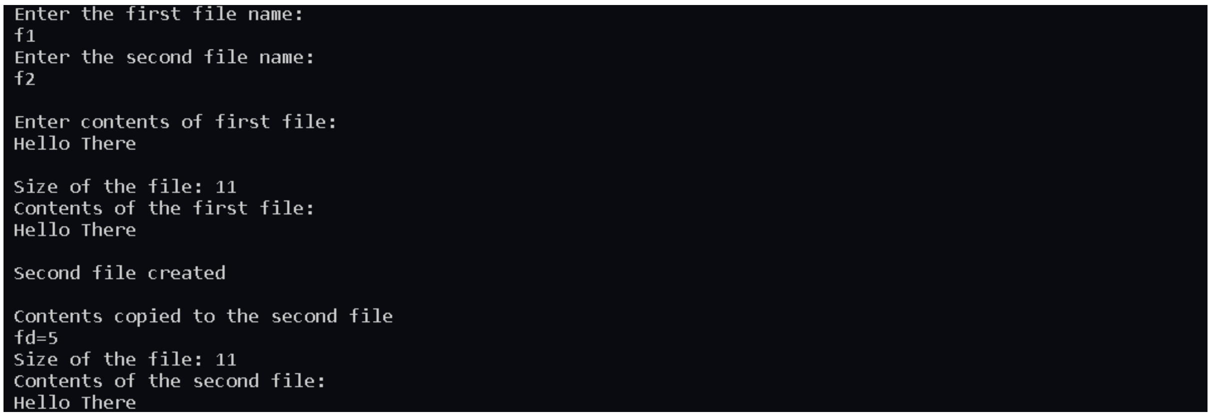


```
Initial file size: 4018
Final file size: 6027
Size of holes:
final size - initial size= 2009 bytes
```

Assignment 2

```
#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<errno.h>
#include<unistd.h>
#include<string.h>
#include<sys/types.h>
#include<sys/stat.h>
int main(){
    int fd1, fd2;
    char c1[300];
    char str[256];
    char file[100];
    char ofile[100];
    size_t sz;

    printf("Enter the first file name: \n");
    scanf("%s", file);
    printf("Enter the second file name: \n");
    scanf("%s", ofile);
    fd1=open(file,O_WRONLY|O_CREAT,0777);
    printf("\nEnter contents of first file:\n");
    scanf("\n%[^\n]s", str);

    sz = write(fd1, str, strlen(str));
    printf("\nSize of the file: %ld \n",sz);

    fd1=open(file,O_RDONLY,0777);
    sz=read(fd1, c1, sizeof(c1));

    c1[sz]='\0';
    printf("Contents of the first file:\n");
    printf("%s\n",c1);

    fd2=open(ofile,O_WRONLY|O_CREAT,0777);
    printf("\nSecond file created\n");

    sz = write(fd2, c1, strlen(c1));
    printf("\nContents copied to the second file\nfd=%d\n",fd2);
    printf("Size of the file: %ld \n",sz);

    fd2=open(ofile,O_RDONLY,0777);
    sz=read(fd2,c1,sizeof(c1));

    c1[sz]='\0';
    printf ("Contents of the second file:\n%s\n",c1);
    close(fd1);
    close(fd2);

    return 0;
}
```

Output



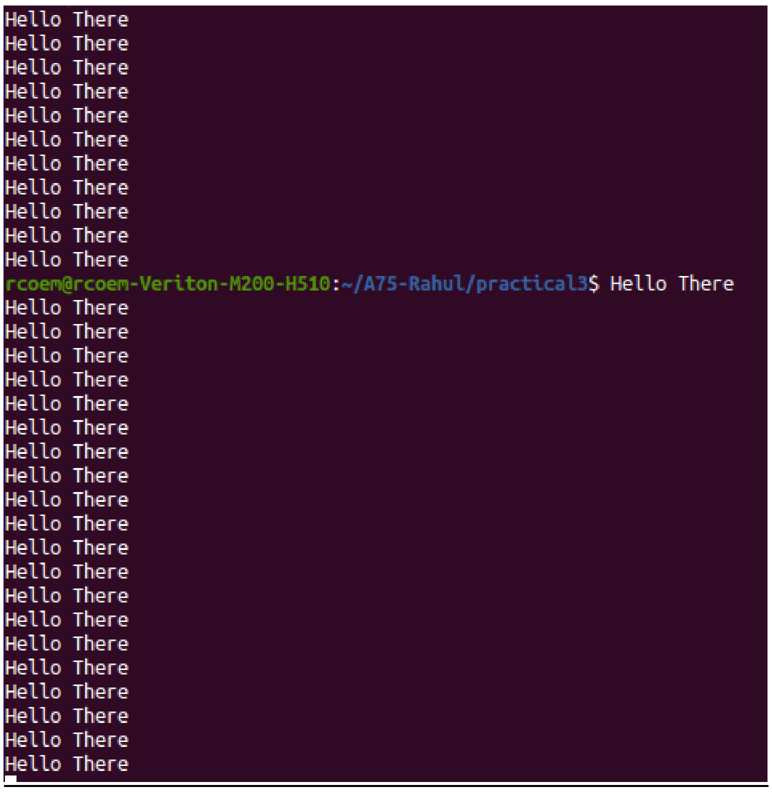
```
Enter the first file name:
f1
Enter the second file name:
f2

Enter contents of first file:
Hello There

Size of the file: 11
Contents of the first file:
Hello There

Second file created

Contents copied to the second file
fd=5
Size of the file: 11
Contents of the second file:
Hello There
```

Code

Program 1

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

Output

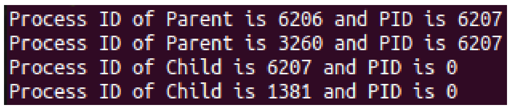


```
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical3$ Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
```

Program 2

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

void forkexample(){
int pid, i, j ;
pid = fork();
	if(pid == 0){
		printf("Process ID of Child is %d and PID is %d",getpid(), pid);
		printf("\n");
		printf("Process ID of Child is %d and PID is %d",getppid(), pid);
		printf("\n");
		//for(i = 0; i < 1000; i++){
		//	printf("Hello from child\n");
		//}
	}
	else{
		printf("Process ID of Parent is %d and PID is %d",getpid(), pid);
		printf("\n");
		printf("Process ID of Parent is %d and PID is %d",getppid(), pid);
		printf("\n");
		//for(j = 0; j < 1000; j++){
		//	printf("Hello from parent :%d",getpid());
		//}
	}
}

int main(){
	forkexample();
	return 0;
}
```

Output

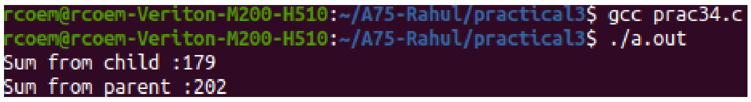


```
Process ID of Parent is 6206 and PID is 6207
Process ID of Parent is 3260 and PID is 6207
Process ID of Child is 6207 and PID is 0
Process ID of Child is 1381 and PID is 0
```

Program 3

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int main(){
	pid_t c;
	int a[10] = {12, 45, 23, 67, 32, 34, 65, 89, 10, 4};
	int sum = 0, i, j;
	if(fork() == 0){
		for(i = 0; i < 5; i++){
			sum += a[i];
		}
		printf("Sum from child :%d\n", sum);
	}
	else{
		c = wait(NULL);
		for(j = 5; j < 10; j++){
			sum += a[j];
		}
		printf("Sum from parent :%d\n", sum);
	}
	return 0;
}
```

Output

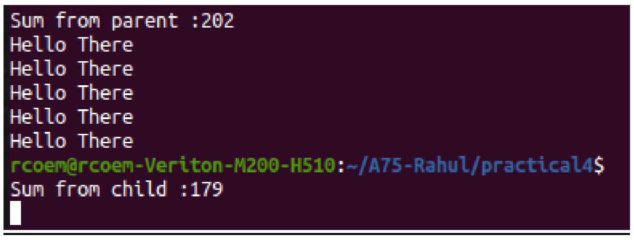


```
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical3$ gcc prac34.c
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical3$ ./a.out
Sum from child :179
Sum from parent :202
```

Program 4

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int main(){
	pid_t c;
	int status;
	int a[10] = {12, 45, 23, 67, 32, 34, 65, 89, 10, 4};
	int sum = 0, i, j;
	if(fork() == 0){
		if(fork() == 0){
			for(i = 0; i < 5; i++){
				printf("Hello There\n");
			}
		}
		else{
			c = waitpid(-1, &status, 0);
			for(i = 0; i < 5; i++){
				sum += a[i];
			}
			printf("\nSum from child :%d\n", sum);
		}
	}
	else{

		for(j = 5; j < 10; j++){
			sum += a[j];
		}
		printf("\nSum from parent :%d\n", sum);
	}
	return 0;
}
```

Output

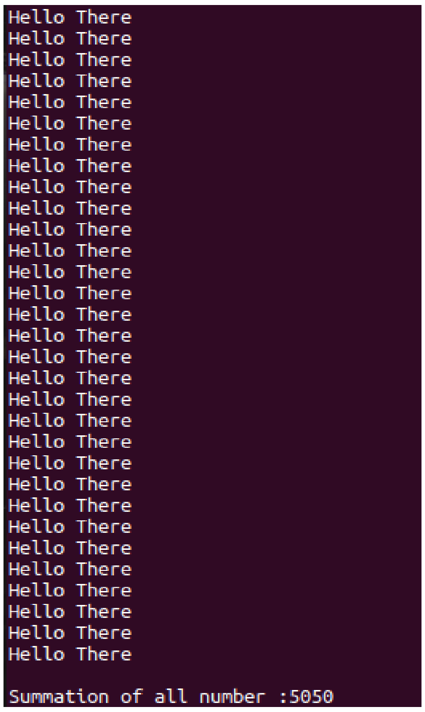


```
Sum from parent :202
Hello There
Hello There
Hello There
Hello There
Hello There
rcoem@rcoem-Veriton-M200-H510:~/A75-Rahul/practical4$
Sum from child :179
```

Program 5

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
int main(){
        pid_t c;
        int status;
        int a[100];
        int sum = 0, i, j;
        if(fork() == 0){
                for(i = 0; i < 100; i++){
                        printf("Hello There\n");
                }
        }
        else{
                c = wait(&status);
                for(i = 0; i < 100; i++){
                        a[i] = (i + 1);
                }
                for(j = 0; j < 100; j++){
                        sum += a[j];
                }
                printf("\nSummation of all number :%d\n", sum);
        }
        return 0;
}
```

Output



```
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There
Hello There

Summation of all number :5050
```