OS-University Practical Examination

Name ; Rahul Goel

Reg No: RA1911030010094

Q1)
a)

```
rahulgoel@Rahuls-MacBook-Air ~ % echo "Current date is `date`"
Current date is Wed Jun  9 10:59:26 IST 2021
rahulgoel@Rahuls-MacBook-Air ~ % echo "User is `who i am`"
User is rahulgoel ttys000  Jun  9 10:59
rahulgoel@Rahuls-MacBook-Air ~ % echo "Current direcotry `pwd`"
Current direcotry /Users/rahulgoel
rahulgoel@Rahuls-MacBook-Air ~ %
```

c)

code:
```c
#include <stdio.h>
int main() {
    int rows, coef = 1, space, i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for (i = 0; i < rows; i++) {
        for (space = 1; space <= rows - i; space++)
            printf("  ");
        for (j = 0; j <= i; j++) {
            if (j == 0 || i == 0)
                coef = 1;
            else
                coef = coef * (i - j + 1) / j;
            printf("%4d", coef);
        }
        printf("\n");
    }
    return 0;
}
```

Q2)

a)

```
rahulgoel@Rahuls-MacBook-Air ~ % cd desktop/
rahulgoel@Rahuls-MacBook-Air desktop % touch pascal.c
rahulgoel@Rahuls-MacBook-Air desktop % gcc pascal.c -o res
rahulgoel@Rahuls-MacBook-Air desktop % ./res
Enter the number of rows: 5
        1
       1  1
      1  2  1
     1  3  3  1
    1  4  6  4  1
rahulgoel@Rahuls-MacBook-Air desktop %
```

code:

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

int main()
{
    int pipefds[2];
    int returnstatus;
    int pid;
    char writemessages[2][20] = {"Hi", "Hello"};
    char readmessage[20];
    returnstatus = pipe(pipefds);
    if (returnstatus == -1)
    {
        printf("Unable to create pipe\n");
```

```c
        return 1;
    }
    pid = fork();

    // Child process
    if (pid == 0)
    {
        read(pipefds[0], readmessage, sizeof(readmessage));
        printf("Child Process - Reading from pipe – Message 1 is %s\n", readmessage);
        read(pipefds[0], readmessage, sizeof(readmessage));
        printf("Child Process - Reading from pipe – Message 2 is %s\n", readmessage);
    }
    else
    { //Parent process
        printf("Parent Process - Writing to pipe - Message 1 is %s\n", writemessages[0]);
        write(pipefds[1], writemessages[0], sizeof(writemessages[0]));
        printf("Parent Process - Writing to pipe - Message 2 is %s\n", writemessages[1]);
        write(pipefds[1], writemessages[1], sizeof(writemessages[1]));
    }
    return 0;
}
```
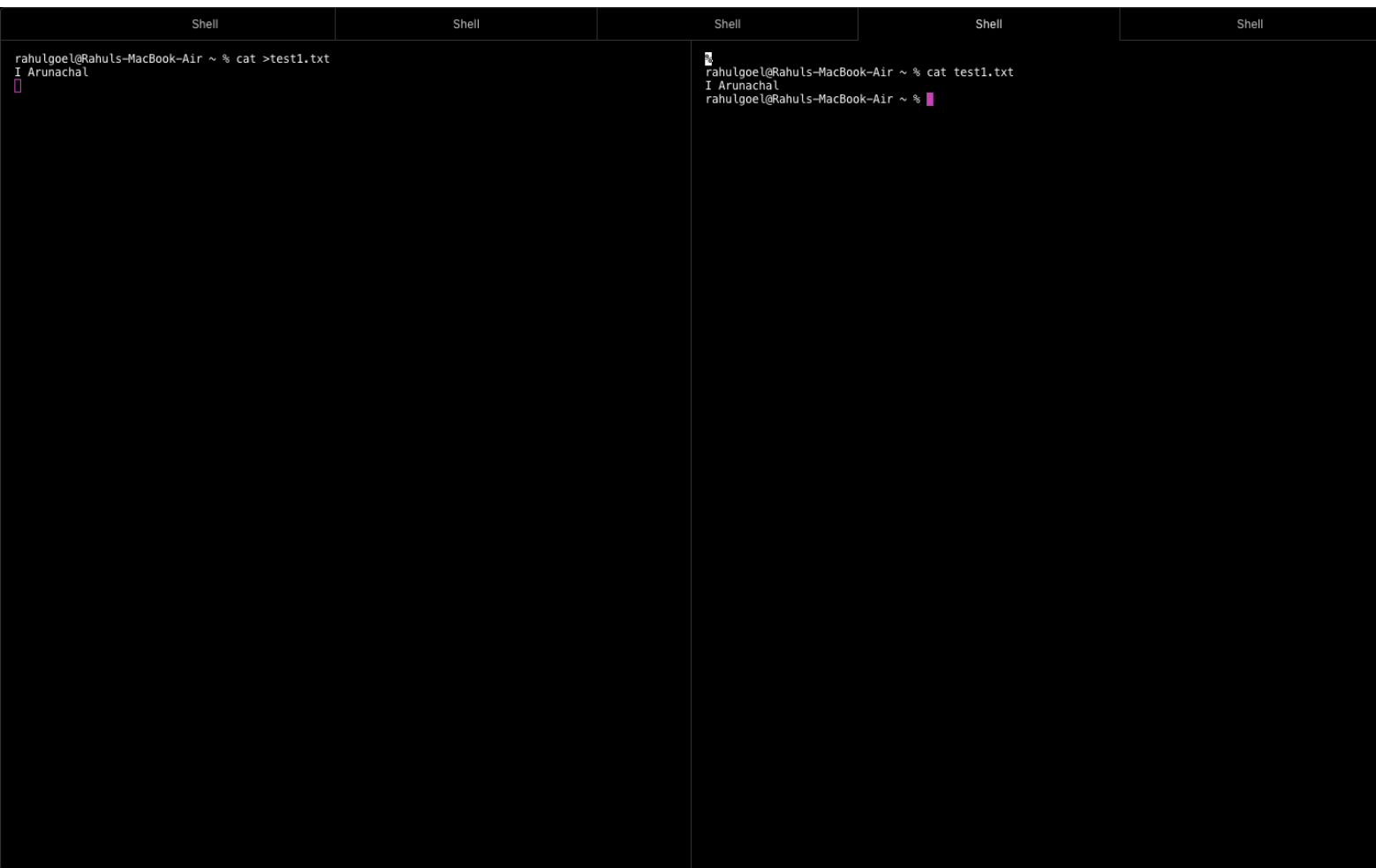
```
rahulgoel@Rahuls-MacBook-Air ~ % cd desktop/
rahulgoel@Rahuls-MacBook-Air desktop % touch pipe.c
rahulgoel@Rahuls-MacBook-Air desktop % gcc pipe.c -o res1
rahulgoel@Rahuls-MacBook-Air desktop % ./res1
Parent Process - Writing to pipe - Message 1 is Hi
Parent Process - Writing to pipe - Message 2 is Hello
Child Process - Reading from pipe - Message 1 is Hi
Child Process - Reading from pipe - Message 2 is Hello
rahulgoel@Rahuls-MacBook-Air desktop %
```

Q1)

b)

```
rahulgoel@Rahuls-MacBook-Air ~ % cat >test1.txt
I Arunachal
```

```
rahulgoel@Rahuls-MacBook-Air ~ % cat test1.txt
I Arunachal
rahulgoel@Rahuls-MacBook-Air ~ %
```

Q2)

b)

code:
file 1

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

int main()
{
    int pp_id;

    pp_id=getpid();


    printf("Process id of example.c= %d\n",pp_id);
    execve("/home/rahul_094/hello",NULL,NULL);

    return 0;
}
```

file 2

```c
#include <stdlib.h>
#include <string.h>
int main(int argc, char *argv[],char *en[])
{
    int i,j,c,ele;
    int arr[argc];

    for (j = 0; j < argc-1; j++)
    {
        int n=atoi(argv[j]);
        arr[j]=n;
    }
    ele=atoi(argv[j]);
    i=0;
    j=argc-1;
```

```c
        c=(i+j)/2;
        while(arr[c]!=ele && i<=j)
        {
            if(ele > arr[c])
                i = c+1;
            else
                j = c-1;

            c = (i+j)/2;
        }
        if(i<=j)
            printf("\nElement Found in the given Array...!!!\n");
        else
            printf("\nElement Not Found in the given Array...!!!\n");
}
```

```
Shell          Shell          Shell          Shell          Shell

rahulgoel@Rahuls-MacBook-Air ~ % cd desktop/
rahulgoel@Rahuls-MacBook-Air desktop % touch file1.c
rahulgoel@Rahuls-MacBook-Air desktop % touch file2.c
rahulgoel@Rahuls-MacBook-Air desktop % chmod 777 file1.c
rahulgoel@Rahuls-MacBook-Air desktop % chmod 777 file2.c
rahulgoel@Rahuls-MacBook-Air desktop % gcc file1.c
rahulgoel@Rahuls-MacBook-Air desktop % gcc -o file2 file2.c
rahulgoel@Rahuls-MacBook-Air desktop % ./a.out
Process id of example.c= 19134
rahulgoel@Rahuls-MacBook-Air desktop %
```