**M Wamiq Akram(20k-1857) Lab 10**

**TASK 1:**

#include<stdio.h>

#include<signal.h>

#include<unistd.h>

void sig\_handler(int signo)

{

if (signo == SIGUSR1)

printf("received SIGUSR1\n");

else if (signo == SIGKILL)

printf("received SIGKILL\n");

else if (signo == SIGSTOP)

printf("received SIGSTOP\n");

}

int main(void){

struct sigaction sa;

sa.sa\_handler=sig\_handler;

if (sigaction(SIGUSR1, &sa,NULL) == -1)

printf("\ncan't catch SIGUSR1\n");

if (sigaction(SIGKILL, &sa,NULL) == -1)

printf("\ncan't catch SIGKILL\n");

if (sigaction(SIGSTOP,&sa,NULL) == -1)

printf("\ncan't catch SIGSTOP\n");

// A long long wait so that we can easily issue a signal to this process

while(1)

sleep(1);

return 0;}

**Task 2:**

#include <stdio.h>

#include <signal.h>

#include <unistd.h>

void sighandler(int sig\_num){

printf("SIGTSTP IGNORE\n");

signal(SIGTSTP, SIG\_IGN);

signal(SIGKILL, SIG\_IGN);

}

int main(){

signal(SIGTSTP, sighandler);

signal(SIGKILL, sighandler);

while(1){

sleep(2);

}

return 0;

}

**Task 3:**

#include<stdio.h>

#include<unistd.h>

#include<wait.h>

#include<signal.h>

int state=1;

void signal\_handler(int NUM){

int num;

if(state==1)

{

printf("Enter Number: ");

scanf("%d",&num);

for(int i=1;i<=10;i++){

printf("%d \* %d = %d\n",num,i,num\*i);}

state=2;

}

else if(state==2)

{

printf("\nIGNORING\n");

signal(SIGINT, SIG\_IGN);

state=3;

}

else if(state==3)

{

int power=num\*num;

printf("\n");

for(int i=1;i<=10;i++){

printf("%d \* %d = %d\n",power,i,power\*i);}

state=4;

}

else if(state==4)

{

printf("\nDEFAULT STATE\n");

signal(SIGINT, SIG\_DFL);

state=5;

}

else if(state==5)

{

signal(SIGINT, SIG\_DFL);

}

}

int main()

{

signal(SIGINT, signal\_handler);

while(1){

sleep(1);

if (state==3)

signal(SIGINT, signal\_handler);

}

return 0;

}