



4) write a shell program to find whether a given year is a leap year or not.

```
echo "Enter a year:"
```

```
read year
```

```
a=$((year % 4))
```

```
b=$((year % 100))
```

```
c=$((year % 400))
```

```
if [ $a -eq 0 -a $b -ne 0 -a $c -eq 0 ]
```

```
then
```

```
    echo "$year is a leap year"
```

```
else
```

```
    echo "$year is not a leap year"
```

```
fi
```

Input: Enter a year : 2020

Output: 2020 is a leap year.



2) write a shell program to find greatest among three numbers.

```
echo "Enter the first number:"
read num1
echo "Enter the second number:"
read num2
echo "Enter the third number:"
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo "$num1 is the greatest number."
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo "$num2 is the greatest number."
else
    echo "$num3 is the greatest number."
fi
```

Input

Enter the first number:

2

Enter the second number:

3

Enter the third number:

5

Output :

5 is the greatest number.



3) write a shell program to display the prime number between 1 and hundred.

```
echo "prime number between 1 and 100 are:"
```

```
for ((i=2; i<=100; i++)); do
```

```
  is_prime=true
```

```
  for ((j=2; j<i; j++)); do
```

```
    if [ $(i%j) -eq 0 ]; then
```

```
      is_prime=false
```

```
      break
```

```
    fi
```

```
done
```

```
if [ "$is_prime" = true ]; then
```

```
  echo $i
```

```
fi
```

```
done
```

Output:

prime numbers between 1 and 100 are :

2	71
3	
5	73
7	79
11	
13	83
17	
19	89
23	
29	97
31	
37	
41	
43	
47	
53	
59	
61	
67	



4) Execute the "ps -ef" and "ps -ux" command, write a shell program that takes the output of these two commands and display the detail about a process whose parent id is 2. The detail about a process will be shown as it is shown during the execution of "ps -ux" command.

Get the output of 'ps -ux' command.

PS-ef_output=\$(ps -ef)

ps-ux_output=\$(ps -ux)

pids=\$(echo "\$ps-ef_output" | awk '\$3==2 {print \$2}')

echo "\$ps-ux_output" | grep -wF "\$pids".

Output

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
kiit	15	0.0	0.0	6072	5152	pts/0	S	21:43	0:00	-bash
kiit	35	0.0	0.0	4776	3264	pts/0	S+	21:47	0:00	bash ritish
kiit	37	0.0	0.0	7784	7784	pts/0	R+	21:47	0:00	ps -ux.



5) write a shell script to extract the date ~~command~~ from the date command, and display the result in numerical form with 12 hour format. (if the date command gives "Thu Jan 2 14:22:54 IST 2014" then display the result in "02/02/2014/2:22:54 PM").

```
formatted_date=$(date "+%d/%m/%Y/%I%P%M:%S%p")
```

```
echo "Formatted Date : $formatted_date"
```

Output

Formatted Date: 06/02/2024/08:34:16 PM



b) Let a directory present in the home directory called "XYZ" that consist of few files and directories. write a shell script to move all the files present in the "XYZ" directory to a subdirectory called "MyFile" and all the subdirectories present in the "XYZ" directory to a subdirectory called "MyDir".

program

```
XYZ_DIR = "$HOME/XYZ"
```

```
MY_FILE_DIR = "$XYZ_DIR/MyFile"
```

```
MY_DIR_DIR = "$XYZ_DIR/MyDir"
```

```
mkdir -p "$MY_FILE_DIR"
```

```
mkdir -p "$MY_DIR_DIR"
```

```
find "$XYZ_DIR" -maxdepth 1 -type f -exec mv {}  
"$MY_FILE_DIR" \;
```

```
find "$XYZ_DIR" -maxdepth 1 -type d -not -path "$MY_DIR_DIR"  
-exec mv {} "$MY_DIR_DIR" \;
```