

Question no 1:

Code:

```
ayesha@student-OptiPlex-7070: ~/Desktop
GNU nano 4.8 Task.sh
read -p "Enter the number:" num
for(( i=1 ; i<5 ; i++ ));do
    echo $num*" "$i"= " $(( $num*$i))
done
```

Output:

```
ayesha@student-OptiPlex-7070:~/Desktop$ nano Task.sh
ayesha@student-OptiPlex-7070:~/Desktop$ ./Task.sh
Enter the number:2
2*1= 2
2*2= 4
2*3= 6
2*4= 8
```

Question 2:

Code :

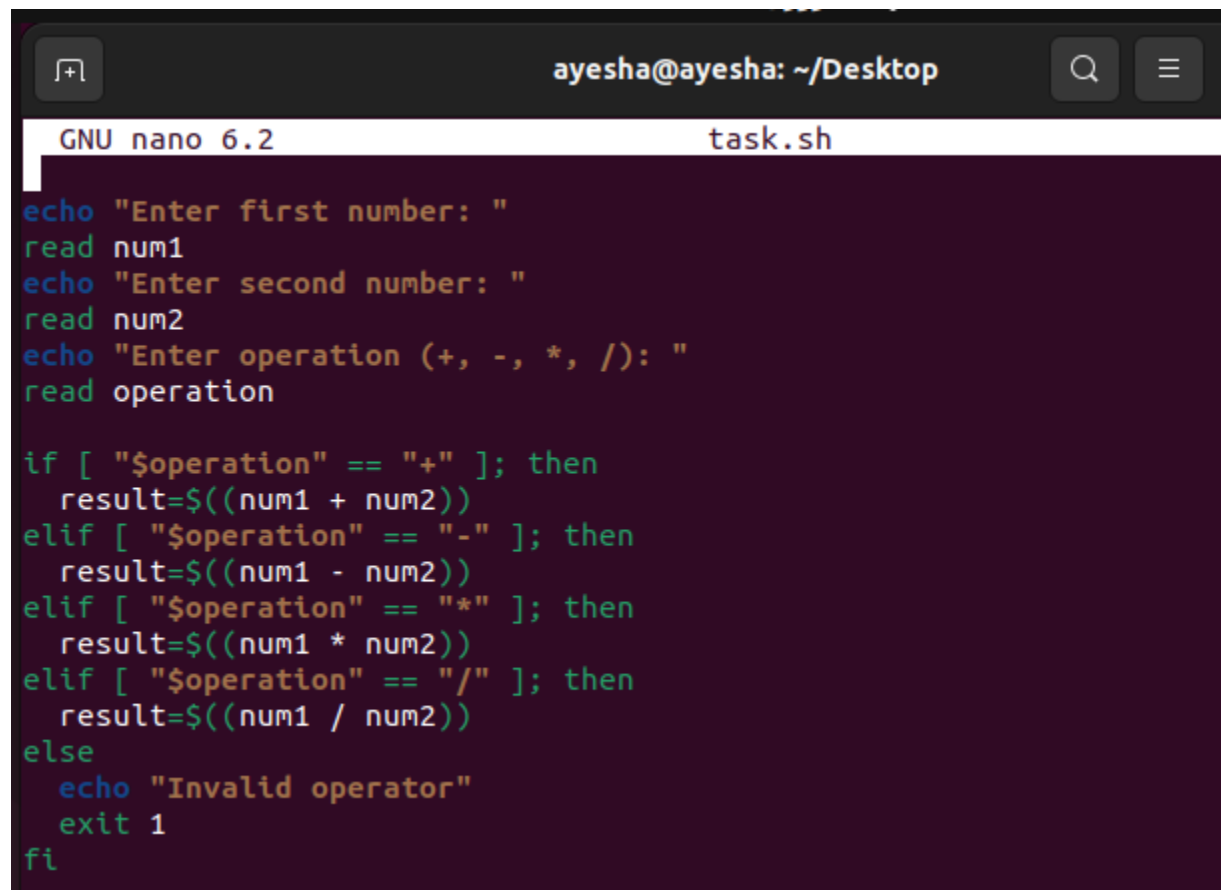
```
ayesha@student-OptiPlex-7070: ~/Desktop
GNU nano 4.8 Task.sh
read -p "Enter the name " name
read -p "Enter the department " depart
read -p "Enter the roll number " rollnum
read -p "Enter the course title " title
echo $name $depart $rollnum $title
```

Output:

```
ayesha@student-OptiPlex-7070:~/Desktop$ nano Task.sh
ayesha@student-OptiPlex-7070:~/Desktop$ ./Task.sh
Enter the name ayesha
Enter the department AI
Enter the roll number 21K4734
Enter the course title Os
ayesha AI 21K4734 Os
```

Question no 3:

Code:



The screenshot shows a terminal window titled "ayesha@ayesha: ~/Desktop". Inside the terminal, the GNU nano 6.2 editor is open, editing a file named "task.sh". The code in the file is a shell script that prompts the user for two numbers and an operation, then performs the operation. The code is as follows:

```
GNU nano 6.2 task.sh
echo "Enter first number: "
read num1
echo "Enter second number: "
read num2
echo "Enter operation (+, -, *, /): "
read operation

if [ "$operation" == "+" ]; then
    result=$((num1 + num2))
elif [ "$operation" == "-" ]; then
    result=$((num1 - num2))
elif [ "$operation" == "*" ]; then
    result=$((num1 * num2))
elif [ "$operation" == "/" ]; then
    result=$((num1 / num2))
else
    echo "Invalid operator"
    exit 1
fi
```

Output :

```
ayesha@ayesha:~/Desktop$ ./task.sh
Enter first number:
5
Enter second number:
4
Enter operation (+, -, *, /):
+
Result: 9
```

Question no 4:

```
GNU nano 6.2 task.sh
cal () {
echo "Enter first number: "
read num1
echo "Enter second number: "
read num2
echo "Enter operation (+, -, *, /): "
read operation

if [ "$operation" == "+" ]; then
    result=$((num1 + num2))
elif [ "$operation" == "-" ]; then
    result=$((num1 - num2))
elif [ "$operation" == "*" ]; then
    result=$((num1 * num2))
elif [ "$operation" == "/" ]; then
    result=$((num1 / num2))
else
    echo "Invalid operator"
    exit 1
fi

echo "Result: $result"
}
cal
```

Output:

```
ayesha@ayesha:~/Desktop$ nano task.sh
ayesha@ayesha:~/Desktop$ ./task.sh
Enter first number:
5
Enter second number:
9
Enter operation (+, -, *, /):
*
Result: 45
```

Question no 5:

Code:

```
echo "Enter the number of terms: "
read n

first=0
second=1

echo "The first $n terms of the Fibonacci series are: "
for ((i=0;i<n;i++))
do
    if [ $i -le 1 ]; then
        fib=$i
    else
        fib=$((first + second))
        first=$second
        second=$fib
    fi
    echo "$fib"
done
```

Output:

```
ayesha@ayesha:~/Desktop$ ./task.sh
Enter the number of terms:
6
The first 6 terms of the Fibonacci series are:
0
1
1
2
3
5
```

Question no 6:

Code

```
mkdir shelldir cdir jpgdir

for file in *
do
    if [ -f "$file" ]; then
        ext="${file##*.}"
        if [ "$ext" == "sh" ]; then
            mv "$file" shelldir
        elif [ "$ext" == "c" ]; then
            mv "$file" cdir
        elif [ "$ext" == "jpg" ]; then
            mv "$file" jpgdir
        fi
    fi
done
```

I don't know how to show its output but I will explain what it is doing :

Explanation:

In this script, the subdirectories `shelldir`, `cdir`, and `jpgdir` are created using the `mkdir` command. Then, the script uses a loop to iterate over all files in the current directory, and checks the extension of each file using parameter expansion. If the extension is `.sh`, the file is moved to the `shelldir` directory, if it is `.c`, it is moved to the `cdir` directory, and if it is `.jpg`, it is moved to the `jpgdir` directory.