

Unix Command Line:

- To copy all files in a particular folder to another folder - cp \$(find input -type f) output/
- To find the students who passed and print only the 1st and 2nd field - grep "PASSED\$" result.csv | cut -d ',' -f 1,2

You have a file named "collect.txt" with the following contents:

sakshamrathi21 1 22b1003@iitb.ac.in submission.sh

malaikaarora01 3 22b0069@iitb.ac.in submsubission.sh

guramritsingh07 2 22b0001@iitb.ac.in submission.sh

rahulgandhi04 5 22b0010@iitb.ac.in subpappu.sh

nithinkamath10 2 zerodha@iitb.ac.in trade.sh

narendramodi24 4 22b0024@iitb.ac.in submission.sh

kavyagupta11 1 22b1053@iitb.ac.in submission.sh

(i) Every line has the following format:

"username<space>number<space>email-id<space>file submitted.

(ii) You need to select the valid lines and print them back into output.txt. The following points will be used to judge whether a line is valid or not.

(iii) Every username contains some characters (of arbitrary length > 0, lowercase letters) in the start and then a two digit number at the end (00 to 99).

(iv) The next number should be between 1 and 4.

(v) The email address should be of the form <22b> <four digit number> <@iitb.ac.in>.

- For the above question, `grep "[a-z]*[0-9][0-9] [1-4] 22b[0-9][0-9][0-9][0-9]@iitb.ac.in submission.sh$" collect.txt > output.txt`

Suppose you have a file "data.txt" with following contents: (X:Y:Z form lines, where X is username, Y is some integer, Z is their place)

sakrat:74:jodhpur

kavgup:95:lucknow

gursin:100:chandigarh

maykum:69:haryana

ridsar:88:jodhpur

You need to use UNIX commands to:

- Sort the lines based on the second column (numeric sorting) and in reverse order. Redirect the output to sort.txt. (Redirection can be done through ">")
- Extract the lines where the third column is "jodhpur". (Perform this on sort.txt and redirect the result to extract.txt)
- Take only the first two columns and print them to display.txt. These columns should be separated by tab instead of colon. (This command has to be performed on extract.txt.)
- Combine these three files and tar them. The final file name will be "submission.tar.gz".

- `sort -nr -t':' -k2,2 data.txt >> sort.txt`
`grep -i "jodhpur" sort.txt >> extract.txt`
`cut extract.txt -d':' -f1,2 --output-delimiter=" " >> display.txt`

```
tar -cvzf submission.tar.gz sort.txt extract.txt display.txt
```

- To change permission to rwx for u and rx for go - `chmod 755 -R input` (or)
`chmod u=rwx -R input ; chmod go=rx -R input`

Makefile:

- `all:main1 main2 main3 main4 main5 main6 main7 main8 main9 comp.o
find.o ops.o add.o rem.o`

```
%.o:src/src[12]/%.cpp
```

```
g++ -c -Wall --std=c++17 -o $@ $^
```

```
main%:main/main%.cpp comp.o find.o ops.o add.o rem.o
```

```
g++ -Wall --std=c++17 -o $@ $^
```

```
clean:
```

```
-rm main*
```

```
rm *.o
```

WebDev:

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="">

  </head>
  <body>

    <script src=""></script>
  </body>
</html>

<nav>
  <ul>
    <li><a href="#Home">Home</a></li>
    <li><a href="#About">About</a></li>
    <li><a href="#Contact">Contact</a></li>
```

```
</ul>
</nav>
```

```
<table>
```

```
  <tr>
```

```
    <th style="text-align:center"><b>Content</b></th>
```

```
    <th style="text-align:center"><b>Weightage</b></th>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Attendance</td>
```

```
    <td>5</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Quizzes</td>
```

```
    <td>20</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Lab Exam</td>
```

```
    <td>60</td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Project</td>
```

```
    <td>15</td>
```

```
  </tr>
```

```
</table>
```

```
<form action="">
```

```
  First name:<input type="text" name="firstname"><br>
```

```
  Email:<input type="email" name="email"><br>
```

```

<input type="number" id="inputNumber" placeholder="Enter a number">
<button onclick="Func()">Find Largest Prime</button>

<!-- Result display -->
<p id="result"></p>

<script>
    function Func() {
        // <!--WRITE YOUR CODE HERE-->
        // YOUR CODE SHOULD WORK, WHEN WE PRESS THE BUTTON, AFTER GIVING A CERTAIN INPUT.
        var x=document.getElementById("inputNumber").value;
        var y=0;
        var i,j,k=0;
        for (i=2;i<=x;i++) {
            k=0;
            for (j=2;j<i;j++) {
                if (i%j==0) {
                    k=1;
                }
            }
            if (k == 0) {
                y=i;
            }
        }
        document.getElementById("result").innerHTML="Largest prime less than or equal to "+x+": "+y;
    }

```

```

<label for="inputN">Enter the value of n:</label>
<input type="number" id="inputN" min="1" step="1" value="1">
<button onclick="Func()">Calculate</button>
<p id="result"></p>

<script>
    function Func() {
        let x=document.getElementById("inputN").value;
        let i,term=1,sum=0,y=2;
        for (y=2;y<=x;y++) {
            term=1;
            for (i=x;i>y;i--) {
                term=term*i;
            }
            sum=sum+Math.pow(-1,y)*term;
        }
        document.getElementById("result").innerHTML="Result: "+sum;
    }

```

```

function makeChanges() {
    let p=document.getElementsByTagName("p");
    for (let i=0;i<p.length;i++) {
        p[i].innerHTML="Enough of Javascript, let's stop.";
    }
    let h1=document.getElementsByTagName("h1");
    for (let t of h1) {
        t.remove();
    }
    let img=document.getElementsByTagName("img");
    for (let i=0;i<img.length;i++) {
        img[i].src="timepass.png";
    }
    let div1=document.getElementsByTagName("div");
    for (let i=0;i<div1.length;i++) {
        if (div1[i].id == "div1") {
            const h3=document.createElement("h3");
            div1[i].appendChild(h3)
        }
    }
    let h2=document.getElementsByTagName("h2");
    for (let i=0;i<h2.length;i++) {
        h2[i].innerHTML=h2[i].innerHTML.toUpperCase();
    }
}

```

```

<form>
  <label for="fullName">Full Name:<input id="fullName" type="text"></label><br><br>
  <label for="email">Email Address:<input id="email" type="email"></label><br><br>
  <label for="password">Password:<input id="password" type="password"></label><br><br>
  <label for="confirmPassword">Confirm Password:<input id="confirmPassword" type="password"></label><br><br>
  <button type="button" id="submit" onclick="validateForm()">Submit</button>
</form>

```

```

function validateName() {
    let name=document.getElementById("fullName").value;
    if (name == "") return true;
    /*Check whether name is entered or not.
    Throw an error if name field is empty.
    Error message will be "Full name is required."*/
}

function validateEmail() {
    let email=document.getElementById("email").value;
    let emailregex=/^[a-z 0-9]+@[a-z]+\.[a-z][a-z][a-z]$/
    if (!emailregex.test(email)) return true;
    /*Check whether email is valid or not, as per the rules stated in problem statement.
    Use regex and test() function to validate the email address.
    Throw an error if email is invalid.
    Error message will be "Invalid Email Address."*/
}

function validatePassword() {
    let pass=document.getElementById("password").value;
    if (pass.length<8) return true;
    /*Check whether password is made of atleast 8 characters.
    /If not, throw an error.
    Error message will be "Password must be at least 8 characters"*/
}

function validateForm() {
    try {

        /*Check whether all fields are entered or not*/
        //Your code here
        if (Checkempty()) throw Error('All fields are required. ');
        if (validateName()) throw Error('Full name is required. ');
        if (validateEmail()) throw Error('Invalid Email Address. ');
        if (validatePassword()) throw Error('Password must contain atleast 8 characters. ');
        if (ConfirmPassword()) throw Error('Passwords do not match. ');
        document.getElementById("feedback").innerHTML='<p style="color:green">'+ "Registration successful!" + "</p>"

        // Additional validation rules can be added here

        //After checking all the rules, if the program throws no error, it will reach this part of code.
        //Using "innerHTML" and "span" tag, give the message "Registration successful!" in GREEN colour to the div container
        "feedback" in index.html.
        //Your code here
    } catch (error) {
        document.getElementById("feedback").innerHTML="<p style='color:red'>"+error+"</p>";
        //After checking all the rules, if the program throws an error, it will reach this part of code.
        //Using "innerHTML" and "span" tag, give the error message in RED colour to the div container "feedback" in index.html.
        //Your code here
    }
}

```

EVALUATE

To add a row to a table, use any of the following 2 methods

```
// var table = document.getElementById("data");  
// var row = table.insertRow();  
// var cell1 = row.insertCell(0);  
// var cell2 = row.insertCell(1);  
// var cell3 = row.insertCell(2);  
// cell1.innerHTML = n_name;  
// cell2.innerHTML = n_email;  
// cell3.innerHTML = n_research;
```

```
const tr = document.createElement('tr');  
let n_name = document.getElementById("fullname").value;  
let n_email = document.getElementById('email').value;  
let n_research = document.getElementById('message').value;  
let tablebody = document.getElementById('data');  
console.log(n_name, n_email, n_research);  
const td1 = document.createElement('td');  
const td2 = document.createElement('td');  
const td3 = document.createElement('td');  
td1.innerHTML = n_name;  
td2.innerHTML = n_email;  
td3.innerHTML = n_research;  
tr.appendChild(td1);  
tr.appendChild(td2);  
tr.appendChild(td3);  
console.log(document.getElementById(data));  
  
tablebody.appendChild(tr);
```