info

ERROR: [2] getimagesize(/home/vol1_7/epizy.com/epiz_32394616/htdocs/cache/SQL_sailer.png): failed to open stream: No such file or dire
/home/vol1_7/epizy.com/epiz_32394616/htdocs/shared/code/tce_functions_tcecode.php

Consider the following database schema to write queries in SQL

image:SQL_sailer.png

i) Find the sailors who have reserved a red boat
ii) Find the names of the sailors who have reserved at least two boats
iii) Find the colors of the boats reserved by ‘Mohan’.
iv) Find the names of sailors who have reserved both a red and a green boat?
v) Find the sids of sailors with age over 20 who have not reserved a red boat.

```
1) select s.name
from sailors s, reserve r, boat b
where s.id = r.sId and r.bId = b.id and b.color = 'red';

2) select s.sid,count(b.bid)
from sailors s, reserves r
where s.sid = r.sid
group by r.sid
having count(r.bid) > = 2 ;

3) SELECT DISTINCT b.color FROM boats b, reserves r, sailors s WHERE s.sname = 'Mohan' AND s.sid = r.sid AND r.bid = b.bid;

4) SELECT s.sname
FROM sailors s, reserve r, boats b
WHERE S.sid = r.sid AND r.bid = b.bid
AND (b.color = ‘red’ OR b.color = ‘green’);

5) SELECT s.sname
FROM sailors s, reserve r, boats b
WHERE b.color != 'red' and b.bid = r.bid and s.sid = r.sid and s.sid > 20;
```

Mayur

Test Execution: Coding

info

Write code for reverse string

```
#include<iostream>
#include<string.h>
using namespace std;
int main ()
{
    char str1[30], temp1;
    int i, j;
    cout << "Enter a string : ";
    gets(str);
    j = strlen(str) - 1;
    for (i = 0; i < j; i++,j--)
    {
        temp1 = str1[i];
        str1[i] = str1[j];
        str1[j] = temp1;
    }
    cout << "\nReverse string : " << str1;
    return 0;
}
```

< previous (1) | (2) confirm | (3) next >

questions

info

Write a code to print all prime numbers between 2 accepted numbers.

```
#include <iostream>
using namespace std;

int main() {

  int l, h, i;
  bool isPrime = true;

  cout << "Enter two numbers : ";
  cin >> l >> h;

  cout << "\nPrime numbers between " << l << " and " << h << " are: " << endl;

  while (l < h) {
    isPrime = true;


    if (l == 0 || l == 1) {
      isPrime = false;
    }

    for (i = 2; i <= l/2; ++i) {
      if (l % i == 0) {
        isPrime = false;
        break;
      }
    }

    if (isPrime)
      cout << low << ", ";

    ++l;
  }

  return 0;
}
```

< previous (5) | (6) confirm | (7) next >

Write a code to check two accepted numbers are amicable or not.

```
#include <bits/stdc++.h>
using namespace std;

int ProDivSum(int n)
{
    int sum = 1;
    for (int i=2; i<=sqrt(n); i++)
    {
        if (n%i == 0)
        {
            sum += i;
            if (n/i != i)
                sum += n/i;
        }
    }
    return sum;
}
bool checkAmicable(int a,int b)
{
    return(ProDivSum(a) == b && ProDivSum(b) == a);
}
int main()
{
    int n, i, j, ctr,nm1,nm2;
    cout << "\n\n Check whether two numbers are Amicable pairs or not: \n";
    cout << "\n Sample: (220, 284), (1184, 1210), (2620, 2924).. \n";
    cout<<"\n";

    cout<<" Input the 1st number : ";
    cin>>nm1;
    cout<<" Input the 2nd number : ";
    cin>>nm2;


    if( checkAmicable(nm1,nm2))
        cout << " The given numbers are an Amicable pair."<<endl;
    else
        cout << " The given numbers are not an Amicable pair."<<endl;
    return 0;
}
```

< previous (8) (9) confirm next >

Test Execution: Coding

info

Write a code to check two accepted numbers are amicable or not.

```
x=int(input('Enter number 1: '))
y=int(input('Enter number 2: '))
sum1=0
sum2=0
for i in range(1,x):
    if x%i==0:
        sum1+=i
for j in range(1,y):
    if y%j==0:
        sum2+=j
if(sum1==y and sum2==x):
    print('Amicable!')
else:
    print('Not Amicable!')
```

< previous (8) | (9) confirm | next >

questions

Test Execution: Coding

info

ERROR: [2] getimagesize(/home/vol1_7/epizy.com/epiz_32394616/htdocs/cache/image_2022-08-20_105516423.png): failed to open /home/vol1_7/epizy.com/epiz_32394616/htdocs/shared/code/tce_functions_tcecode.php

image:image_2022-08-20_105516423.png

Consider the above table :
1. Create this table with all the data as shown in the picture
2. Display details of product whose warranty is more than 2 years
3. Display all the details of Smart Tv
4. Display the price of those products whose quantity is null
5. Display all the products where the price is higher than the average

```
1) create table Product(
    int p_id,
    big int p_price,
    varchar(50) p_name,
    int p_warranty,
    int p_qty
);

2) Select * from PRODUCT where p_warranty > 2 ;
3) select* from Product  where id=2;
4) select * from Product where p_qty=null;

5)
```