

Payoda_Phase2

MongoDB & C#

Task 1:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.3
Microsoft Windows [Version 10.0.22631.2115]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sanjai>mongosh
Current Mongosh Log ID: 64cb3079de68459c1ebd955a
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.3
Using MongoDB:      6.0.8
Using Mongosh:      1.10.3

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2023-08-02T15:45:04.388+05:30: Access control is not enabled for the database. Read and write access to data and configuration is
  unrestricted
  -----

test> show dbs
Crud          72.00 KiB
FlightReservation 252.00 KiB
admin         40.00 KiB
config        60.00 KiB
local         88.00 KiB
myapp         40.00 KiB
test> show dbs
Crud          72.00 KiB
FlightReservation 252.00 KiB
Sample        8.00 KiB
admin         40.00 KiB
config        72.00 KiB
local         88.00 KiB
myapp         40.00 KiB
test> db;
test
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.3
test> use payoda
switched to db payoda
payoda> db.createCollection("Employees");
{ ok: 1 }
payoda> db.Employees.insertOne({name:Sanjai})
ReferenceError: Sanjai is not defined
payoda> db.Employees.insertOne({name:"Sanjai"})
{
  acknowledged: true,
  insertedId: ObjectId("64cb3d63de68459c1ebd955b")
}
payoda> db.Employees.insertMany({name:"Sanjai"},{name:"kumar"})
MongoInvalidArgumentError: Argument "docs" must be an array of documents
payoda> db.Employees.insertMany([ {name:"Sanjai"}, {name:"kumar"} ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64cb3dfade68459c1ebd955c"),
    '1': ObjectId("64cb3dfade68459c1ebd955d")
  }
}
payoda> db.Employees.find()
[
  { _id: ObjectId("64cb3d63de68459c1ebd955b"), name: 'Sanjai' },
  { _id: ObjectId("64cb3dfade68459c1ebd955c"), name: 'Sanjai' },
  { _id: ObjectId("64cb3dfade68459c1ebd955d"), name: 'kumar' }
]
payoda> db.Employees.find({}, {name:0})
[
  { _id: ObjectId("64cb3d63de68459c1ebd955b") },
  { _id: ObjectId("64cb3dfade68459c1ebd955c") },
  { _id: ObjectId("64cb3dfade68459c1ebd955d") }
]
payoda> db.Employees.find({}, {name:1})
```

```
mongosh mongodb://127.0.0.1:27017/
payoda> db.Employees.find({}, {name:1})
[
  { _id: ObjectId("64cb3d63de68459c1ebd955b"), name: 'Sanjai' },
  { _id: ObjectId("64cb3dfade68459c1ebd955c"), name: 'Sanjai' },
  { _id: ObjectId("64cb3dfade68459c1ebd955d"), name: 'kumar' }
]
payoda> db.users.insertOne({})
{
  acknowledged: true,
  insertedId: ObjectId("64cb4cd5de68459c1ebd955e")
}
payoda> db.users.find()
[ { _id: ObjectId("64cb4cd5de68459c1ebd955e") } ]
payoda> db.users.insertOne({name:"Sanjai",age:22,email:sanjai@gmail.com})
Uncaught:
SyntaxError: Unexpected token, expected ",", (1:53)

> 1 | db.users.insertOne({name:"Sanjai",age:22,email:sanjai@gmail.com})
    |                                     ^
    2 |

payoda> db.users.insertOne({name:"Sanjai",age:22,email:"sanjai@gmail.com"})
{
  acknowledged: true,
  insertedId: ObjectId("64cb4d4dde68459c1ebd955f")
}
payoda> db.users.find()
[
  { _id: ObjectId("64cb4cd5de68459c1ebd955e") },
  {
    _id: ObjectId("64cb4d4dde68459c1ebd955f"),
    name: 'Sanjai',
    age: 22,
    email: 'sanjai@gmail.com'
  }
]
```

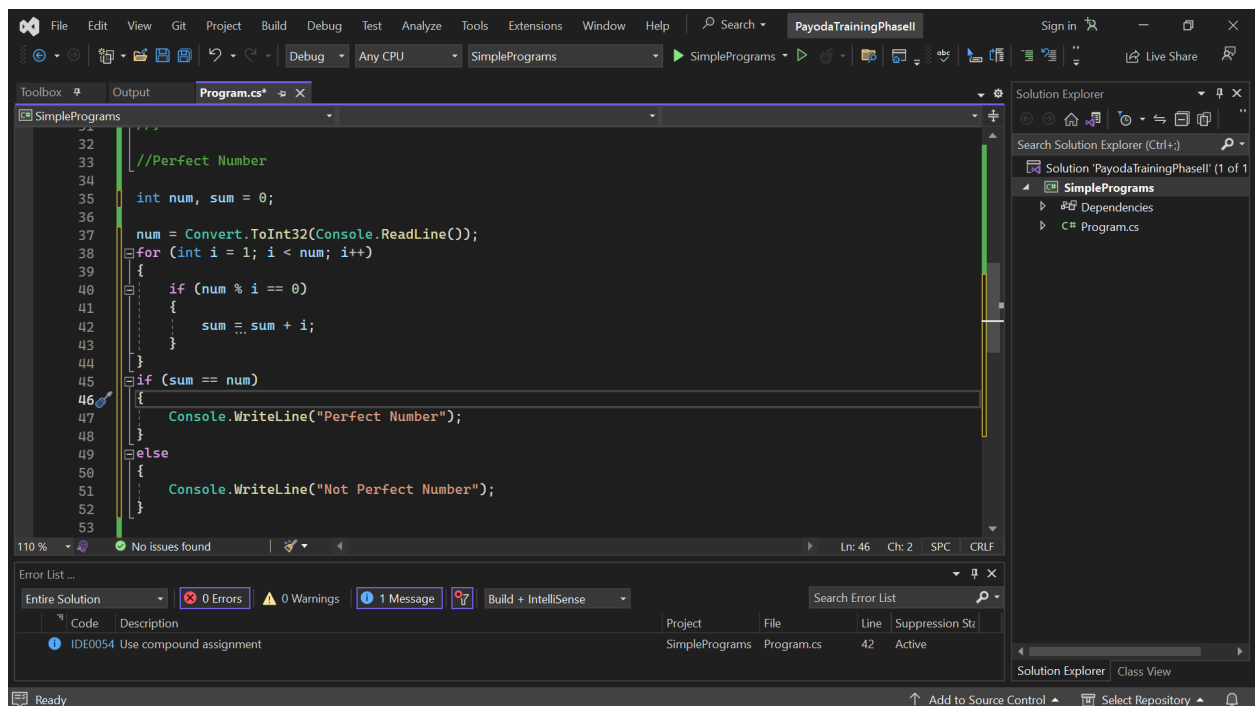
```
Uncaught:
SyntaxError: Unexpected token, expected ",", (1:53)

> 1 | db.users.insertOne({name:"Sanjai",age:22,email:sanjai@gmail.com})
    |                                     ^
    2 |

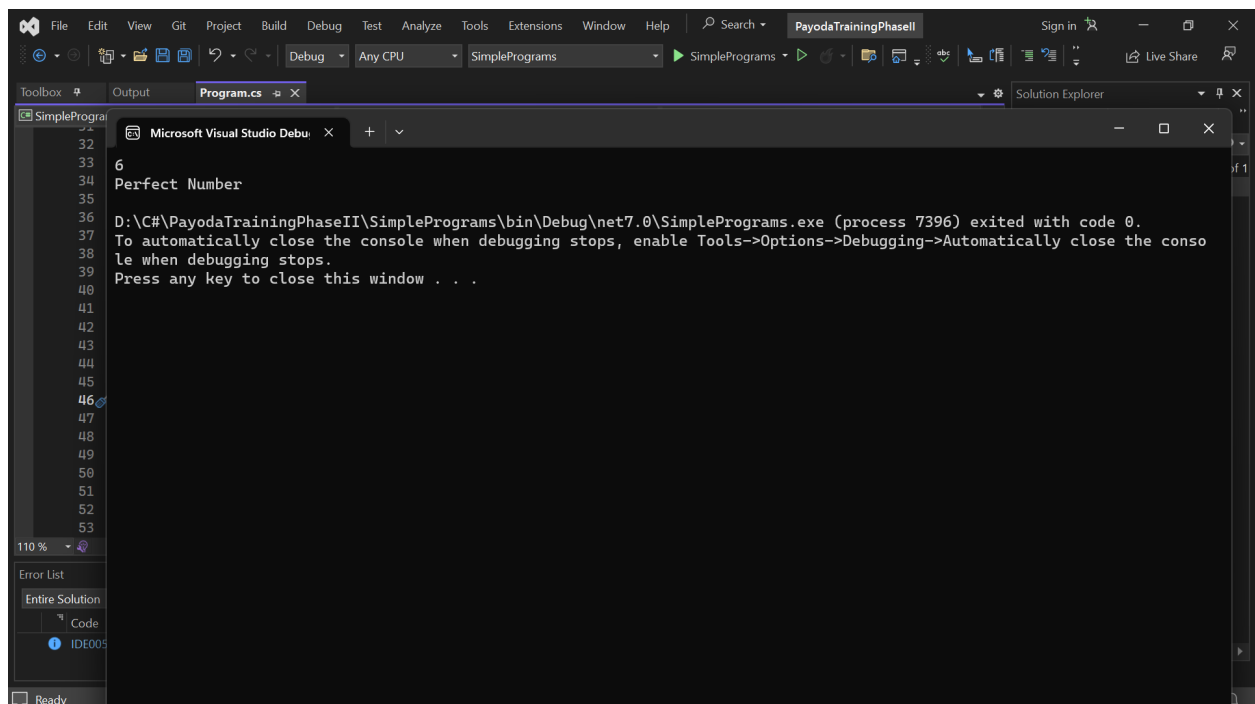
payoda> db.users.insertOne({name:"Sanjai",age:22,email:"sanjai@gmail.com"})
{
  acknowledged: true,
  insertedId: ObjectId("64cb4d4dde68459c1ebd955f")
}
payoda> db.users.find()
[
  { _id: ObjectId("64cb4cd5de68459c1ebd955e") },
  {
    _id: ObjectId("64cb4d4dde68459c1ebd955f"),
    name: 'Sanjai',
    age: 22,
    email: 'sanjai@gmail.com'
  }
]
payoda> db.users.find({age:{$lt:25}})
[
  {
    _id: ObjectId("64cb4d4dde68459c1ebd955f"),
    name: 'Sanjai',
    age: 22,
    email: 'sanjai@gmail.com'
  }
]
payoda> db.users.find({age:{$gt:25}})
[]
payoda>
```

Task 2:

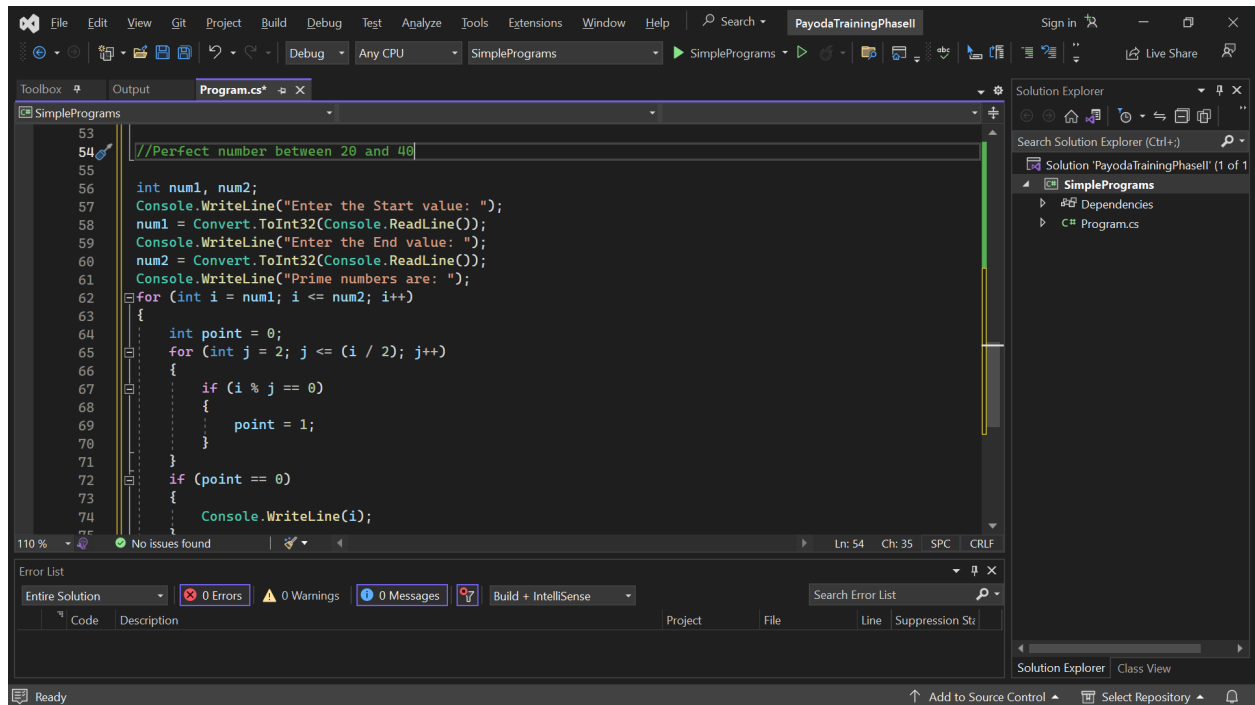
Perfect Number:



Output:



Prime number between 20 and 40:

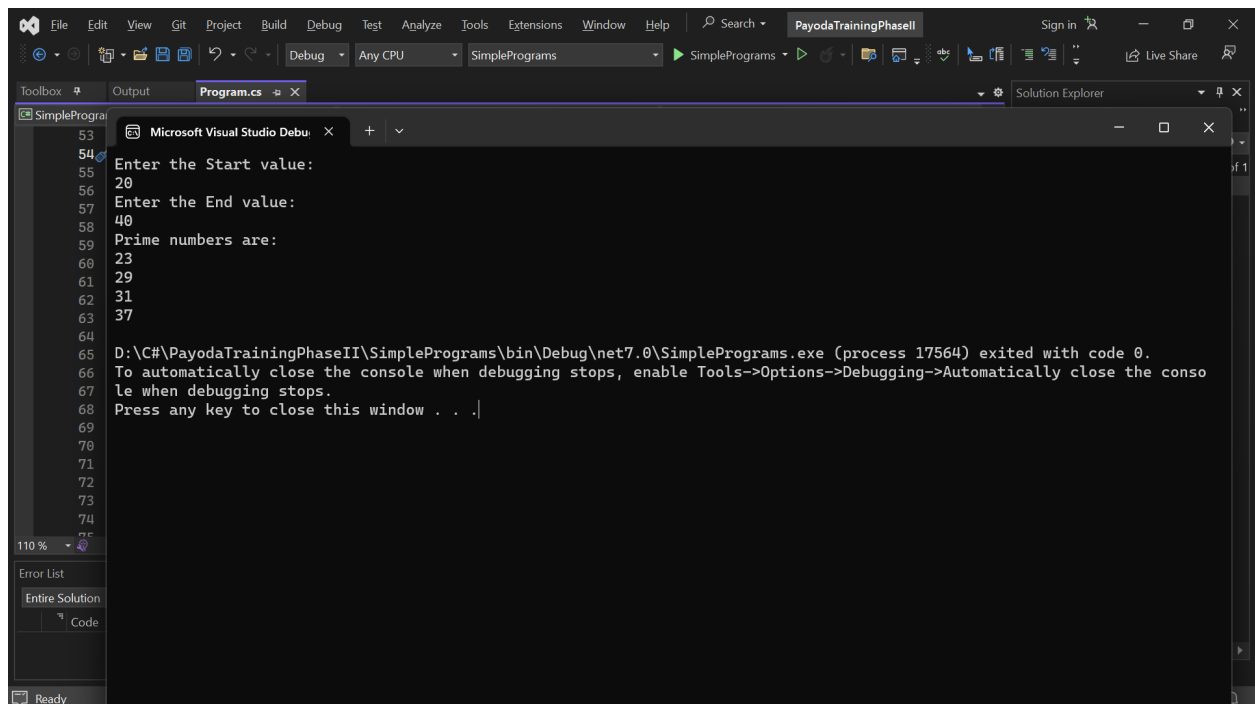


The screenshot shows the Visual Studio IDE with a C# program named 'SimplePrograms'. The code is as follows:

```
53
54 //Perfect number between 20 and 40
55
56 int num1, num2;
57 Console.WriteLine("Enter the Start value: ");
58 num1 = Convert.ToInt32(Console.ReadLine());
59 Console.WriteLine("Enter the End value: ");
60 num2 = Convert.ToInt32(Console.ReadLine());
61 Console.WriteLine("Prime numbers are: ");
62 for (int i = num1; i <= num2; i++)
63 {
64     int point = 0;
65     for (int j = 2; j <= (i / 2); j++)
66     {
67         if (i % j == 0)
68         {
69             point = 1;
70         }
71     }
72     if (point == 0)
73     {
74         Console.WriteLine(i);
75     }
76 }
```

The Solution Explorer on the right shows the project structure: 'SimplePrograms' containing 'Dependencies' and 'Program.cs'. The status bar at the bottom indicates 'No issues found'.

Output:



The screenshot shows the Visual Studio IDE with the 'Output' window open, displaying the execution of the program. The output is as follows:

```
53
54 Enter the Start value:
55 20
56 Enter the End value:
57 40
58 Prime numbers are:
59 23
60 29
61 31
62 37
63
64
65 D:\C#\PayodaTrainingPhaseII\SimplePrograms\bin\Debug\net7.0\SimplePrograms.exe (process 17564) exited with code 0.
66 To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
67 Press any key to close this window . . .|
68
69
70
71
72
73
74
```

The 'Error List' window at the bottom left shows '0 Errors' and '0 Warnings'. The status bar at the bottom indicates 'Ready'.

