[CSE 131s]

Computer Programming

TASK 3

Capstone Project

Name |

OMAR ASHRAF ABDELSATAR AHMED

ID |

2100354

Preview

```cpp
#include <iostream>
using namespace std;
double totalRes(string des, char connection);

int main()
{
  // Getting Circuit Description From User
  string des;
  cout << " Circuit Description : ";
  getline(cin, des);

  // Getting Voltage Applied From User
  float V;
  cout << " Voltage Applied = ";
  cin >> V;

  // Validating User Input
  bool notValidInput = false;
  for (int i = 0; i < des.length(); i++)
  {
    switch (des[i])
    {
    case 'S': case 'P': case 'e': case 'E':
    case '.': case ' ': case '0' ... '9':
    break;
    default: notValidInput = true;
    }
  }
  if (notValidInput == true)
    cout << "Wrong Circuit Description" << endl;

  // Returning Output to The User
  if (notValidInput == false)
  {
    double Req = totalRes(des, des[0]);
    cout << " Req = " << Req << " ohm" << endl;
    cout << " I = " << V / Req << " Amp" << endl;
  }
}

double totalRes(string des, char connection)
{
  double Req = 0, R;
  int childConnection = 0, digit = 0, indexFirstCon;
  // Main for Loop
  for (int i = 2; i < des.length(); i++)
  {
    // Checking for internal connections
    if (des[i] == 'S' || des[i] == 'P')
    {
      childConnection = 1;
      indexFirstCon = i;
    }
```

```
    // No internal Connection ==== > like Task 2
    if (childConnection == 0)
    {
      if (des[i] == ' ')
      {
        R = stof(des.substr(i - digit, digit));
        switch (des[0])
        {
        case 'S':
          Req += R;
          break;
        case 'P':
          Req += 1 / R;
          break;
        }
        digit = -1;
      }
      digit++;
    }

    // There is Internal Connection ==== > Task 3
    if (childConnection == 1)
    {
      if (des[i] == 'e')
      {
        string branch = des.substr(indexFirstCon, i -
indexFirstCon + 1);
        R = totalRes(branch, branch[0]);
        switch (connection)
        {
        case 'S':
          Req += R;
          break;
        case 'P':
          Req += 1 / R;
          break;
        }
        childConnection = 0;
        i++;
      }
    }
  }
  // Giving The Results to the user
  switch (connection)
  {
  case 'S':
  case 's':
    return Req;
  case 'P':
  case 'p':
    return 1 / Req;
  }
  return 0;
}
```

Source Code

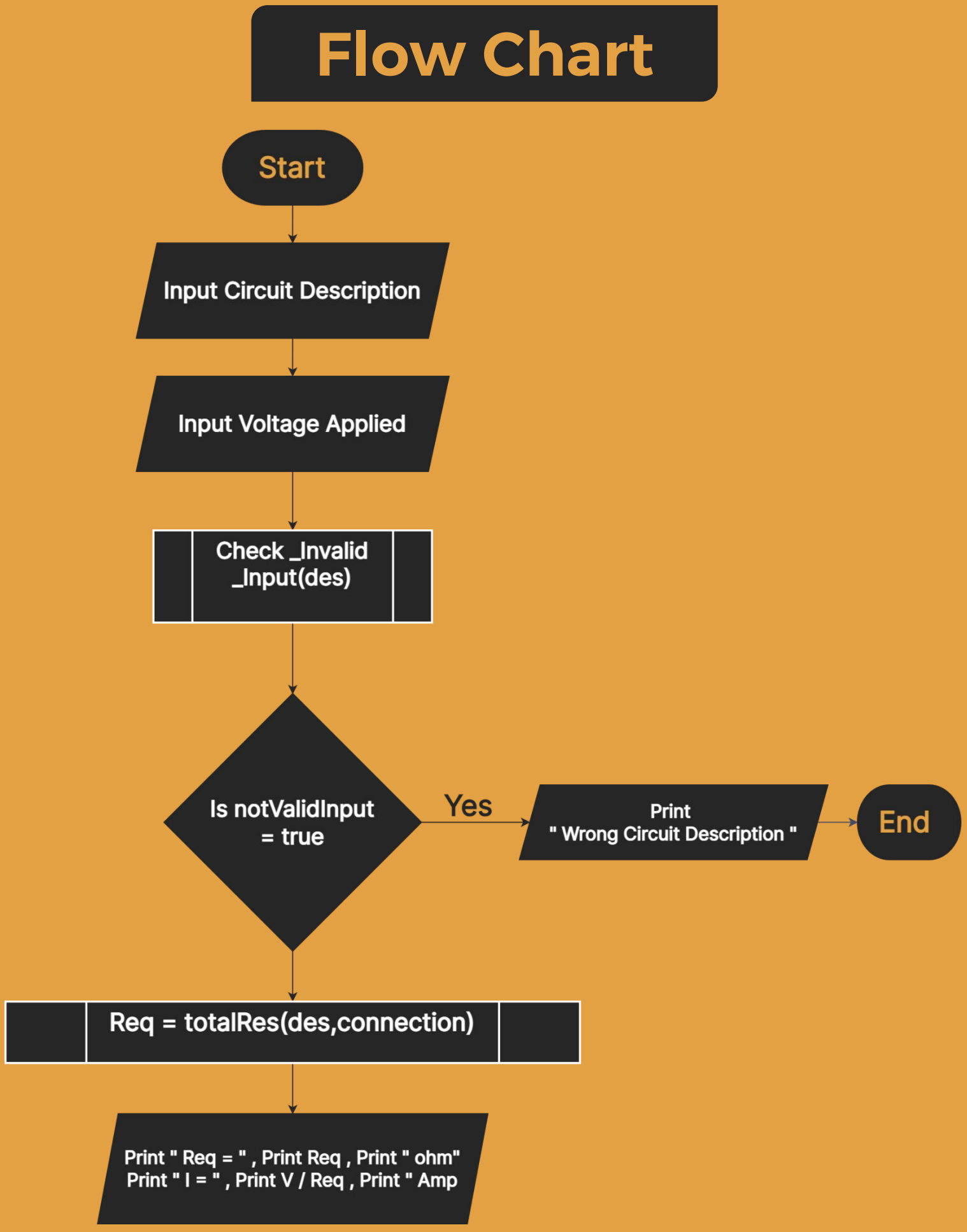
1

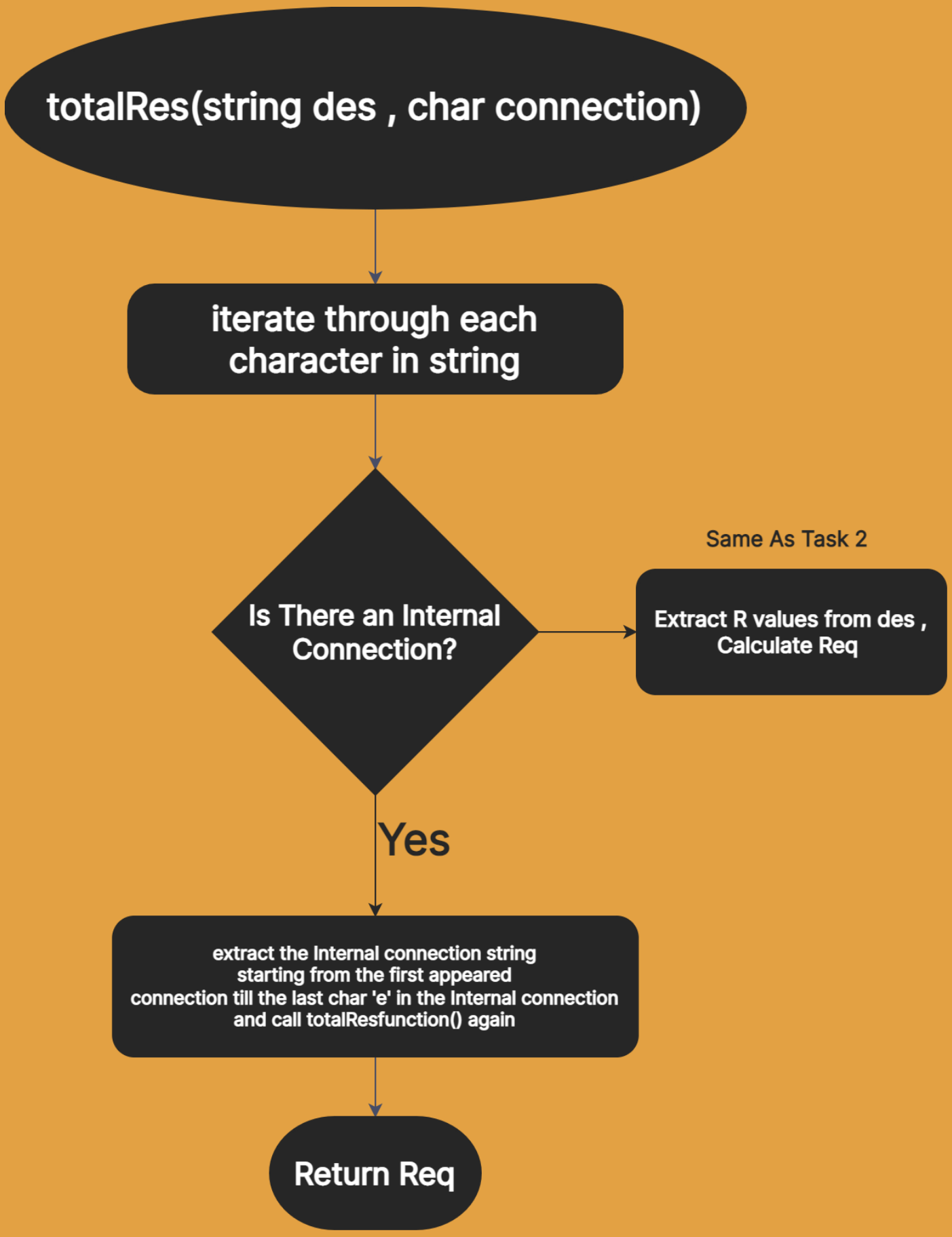
```
#include <iostream>
using namespace std;
double totalRes(string des, char connection);
int main()
{
  // Getting Circuit Description From User
  string des;
  cout << " Circuit Description : ";
  getline(cin, des);
  // Getting Voltage Applied From User
  float V;
  cout << " Voltage Applied = ";
  cin >> V;
  // Validating User Input
  bool notValidInput = false;
  for (int i = 0; i < des.length(); i++)
  {
    switch (des[i])
    {     case 'S': case 'P': case 'e': case 'E': case '.': case ' ': case '0' ... '9':
    break;
    default: notValidInput = true;  }
  }
  // Returning Output to The User
  if (notValidInput == false)
  {
    double Req = totalRes(des, des[0]);
    cout << " Req = " << Req << " ohm" << endl;
    cout << " I = " << V / Req << " Amp" << endl;
  }
}
```

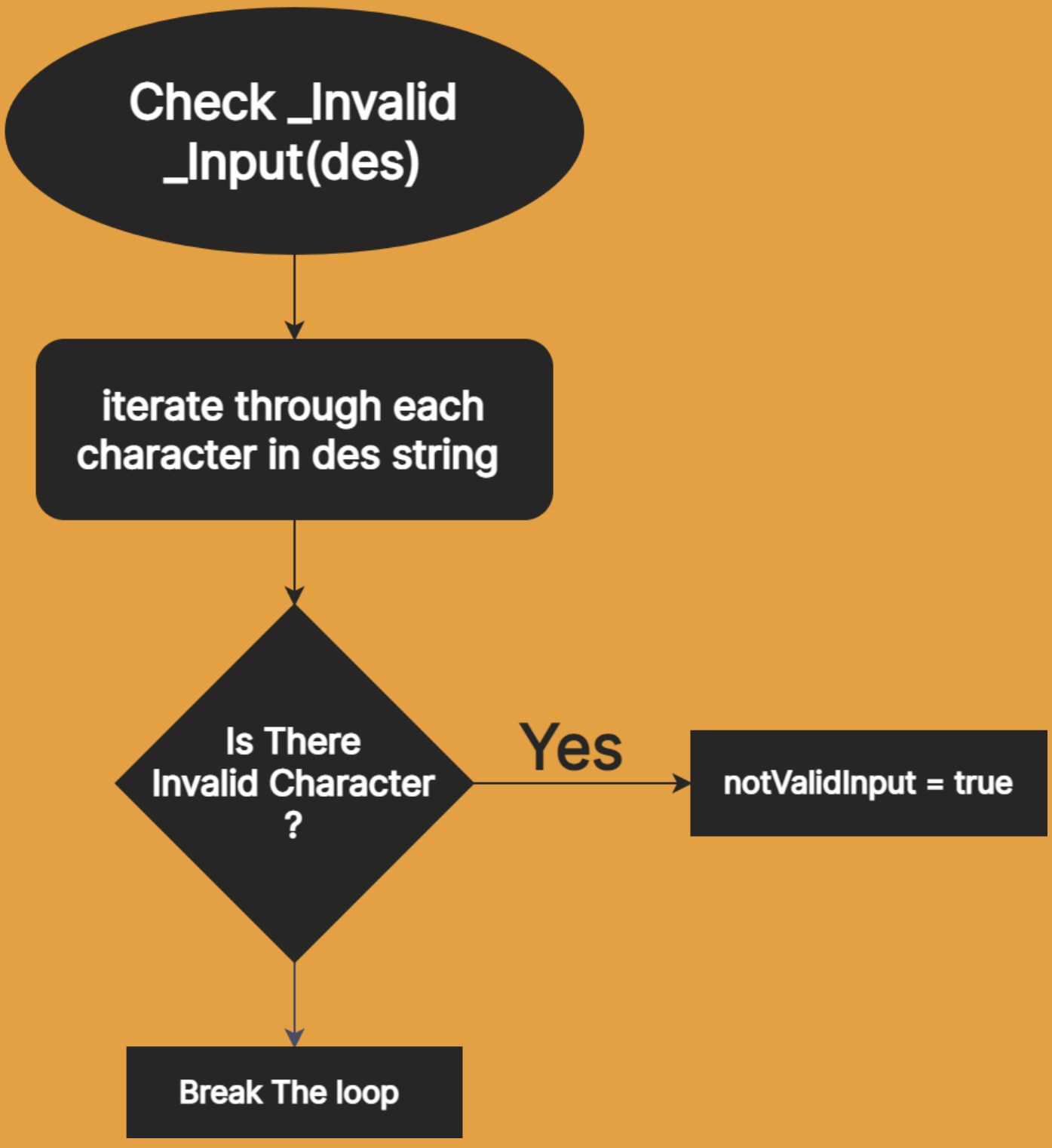
```
double totalRes(string des, char connection)
{
  double Req = 0, R;
  int childConnection = 0, digit = 0, indexFirstCon;
  // Main for Loop
  for (int i = 2; i < des.length(); i++)
  {
    // Checking for internal connections
    if (des[i] == 'S' || des[i] == 'P')
    {
      childConnection = 1;
      indexFirstCon = i;
    }

    // No internal Connection ====> like Task 2
    if (childConnection == 0)
    {
      if (des[i] == ' ')
      {
        R = stof(des.substr(i - digit, digit));
        switch (des[0])
        {
        case 'S':
          Req += R;
          break;
        case 'P':
          Req += 1 / R;
          break;
        }
        digit = -1;
      }
      digit++;
    }
```

```
  // There is Internal Connection ==== > Task 3
  if (childConnection == 1)
  {
    if (des[i] == 'e')
    {
      string branch = des.substr(indexFirstCon, i - indexFirstCon + 1);
      R = totalRes(branch, branch[0]);
      switch (connection)
      {
      case 'S':
        Req += R;
        break;
      case 'P':
        Req += 1 / R;
        break;
      }
      childConnection = 0;
      i++;
    }
  }
 }
 // Giving The Results to the user
 switch (connection)
 {
 case 'S':
 case 's':
   return Req;
 case 'P':
 case 'p':
   return 1 / Req;
 }
 return 0;
}
```

Flow Chart


Start
Input Circuit Description
Input Voltage Applied
Check _Invalid _Input(des)
Is notValidInput = true
Yes
Print " Wrong Circuit Description "
End
Req = totalRes(des,connection)
Print " Req = " , Print Req , Print " ohm"
Print " I = " , Print V / Req , Print " Amp


totalRes(string des , char connection)
iterate through each character in string
Is There an Internal Connection?
Same As Task 2
Extract R values from des , Calculate Req
Yes
extract the Internal connection string starting from the first appeared connection till the last char 'e' in the Internal connection and call totalResfunction() again
Return Req


Check _Invalid _Input(des)
iterate through each character in des string
Is There Invalid Character ?
Yes
notValidInput = true
Break The loop

Test Cases

1

```
Circuit Description : S 1.5 P 12.85 3.6 e P 5 6.6 e 7 E
Voltage Applied = 3.8
Req = 14.157 ohm
I = 0.268419 Amp
```

2

```
Circuit Description : S L 2.5 5.2 e 4.7 8 E
Voltage Applied = 9
Wrong Circuit Description
```

3

```
Circuit Description : P S 4.7 4.7 e 4.7 S 4.7 4.7 e E
Voltage Applied = 7
Req = 2.35 ohm
I = 2.97872 Amp
```

4

```
Circuit Description : P S 4.7 4.7 e 4.7 4.7 E
Voltage Applied = 9
Req = 1.88 ohm
I = 4.78723 Amp
```

5

```
Circuit Description : Z S 8.2 3.1 e 1.3 7.8 E
Voltage Applied = 5
Wrong Circuit Description
```

6

```
Circuit Description : P S 8.2 3.1 e S 1.3 7.8 e E
Voltage Applied = 5
Req = 5.04069 ohm
I = 0.991928 Amp
```