ज्ञानं परमं भूषणम्

SILVER OAK UNIVERSITY

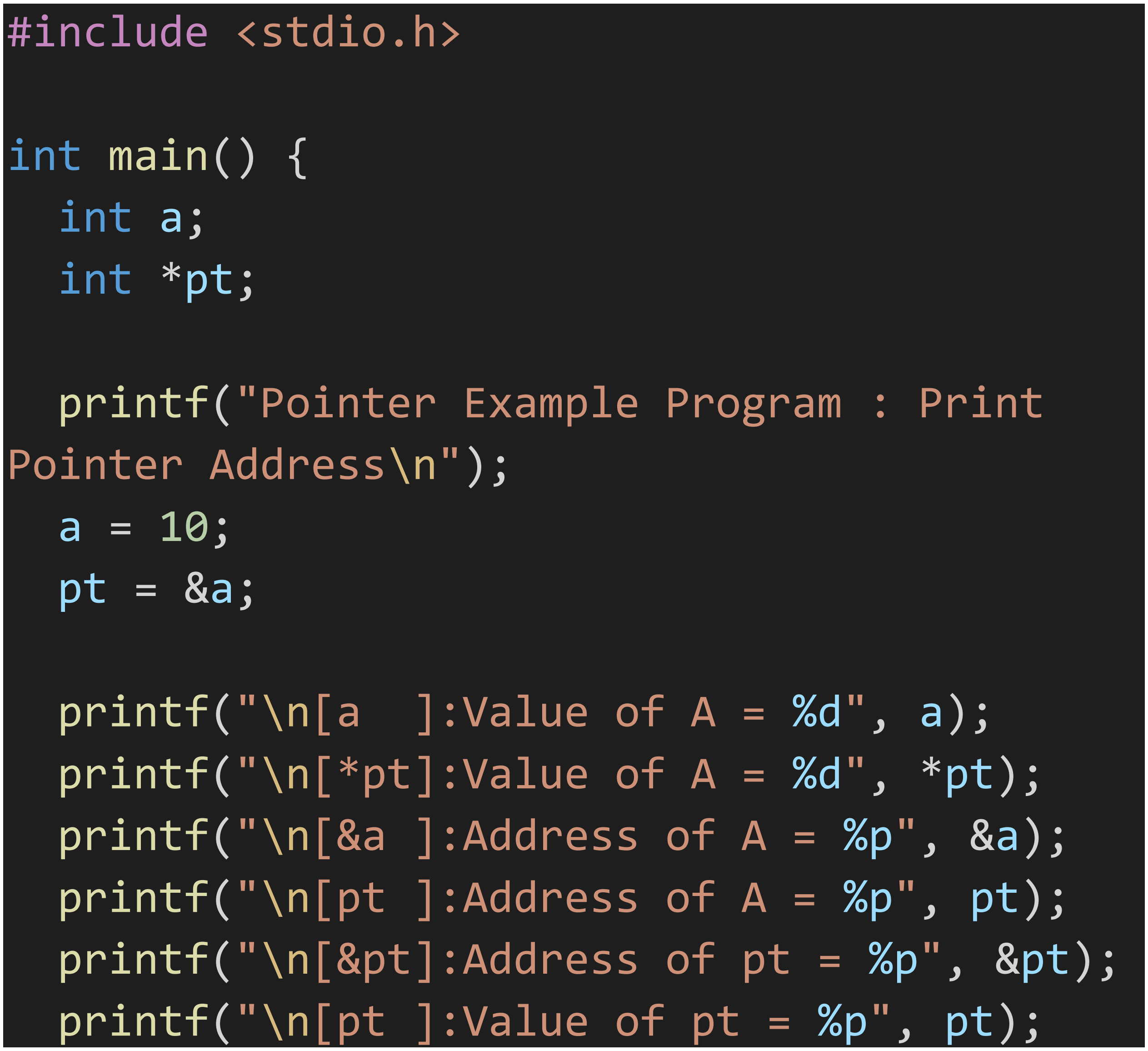
EDUCATION TO INNOVATION

NAME	PATEL RITESH ALPESHBHAI
EN. NO.	2202031000114
COURSE	B. TECH IT
SUBJECT	OJT PRACTICALS

Practical :- 1

AIM :- Write a program to print the address of a variable using a pointer.

Code :-



```c
#include <stdio.h>

int main() {
  int a;
  int *pt;

  printf("Pointer Example Program : Print Pointer Address\n");
  a = 10;
  pt = &a;

  printf("\n[a  ]:Value of A = %d", a);
  printf("\n[*pt]:Value of A = %d", *pt);
  printf("\n[&a ]:Address of A = %p", &a);
  printf("\n[pt ]:Address of A = %p", pt);
  printf("\n[&pt]:Address of pt = %p", &pt);
  printf("\n[pt ]:Value of pt = %p", pt);

  return 0;
}
```

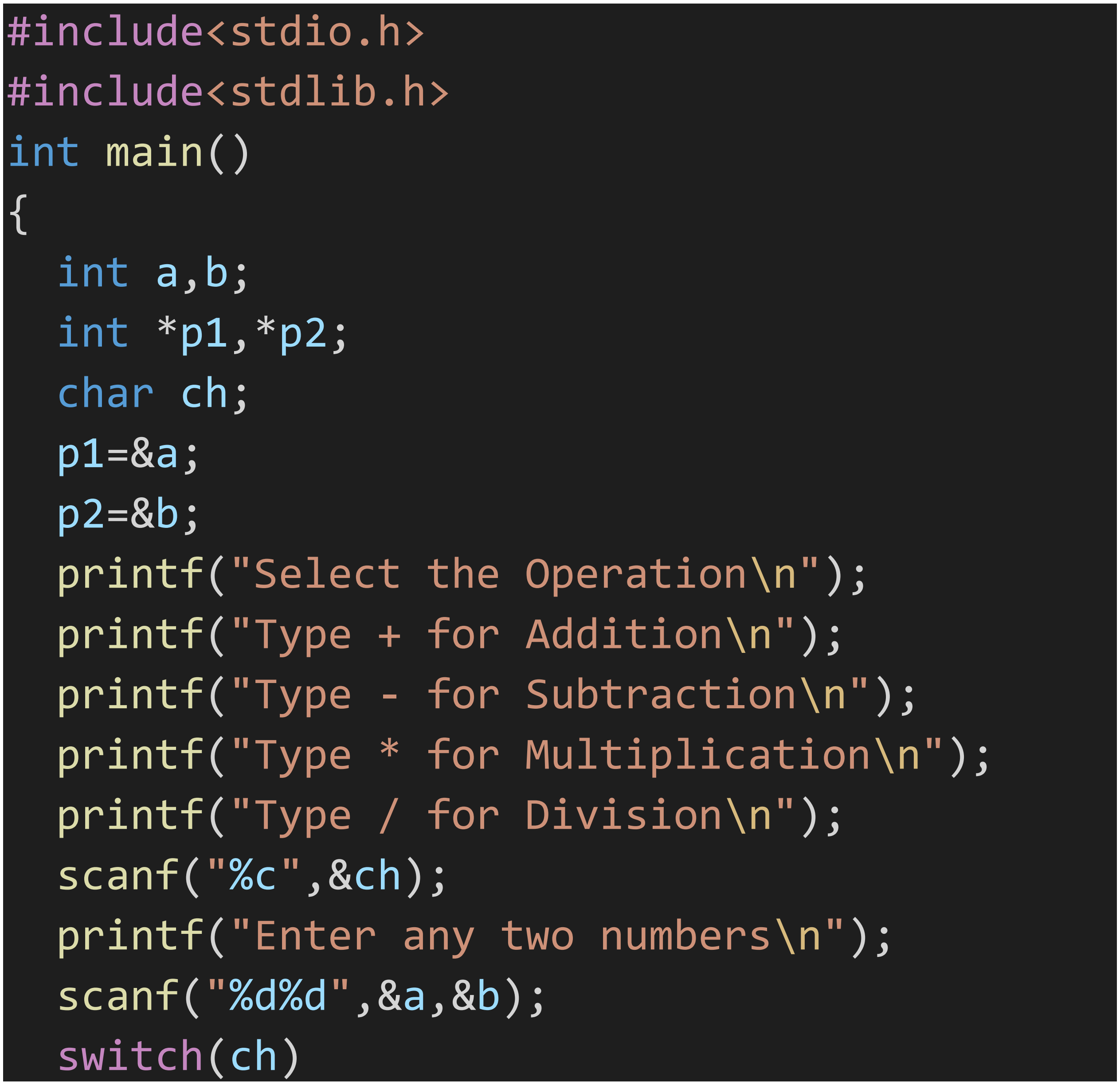
Output :-

```
Pointer Example Program : Print Pointer
Address

[a  ]:Value of A = 10
[*pt]:Value of A = 10
[&a ]:Address of A = 0060FF0C
[pt ]:Address of A = 0060FF0C
[&pt]:Address of pt = 0060FF08
[pt ]:Value of pt = 0060FF0C
```

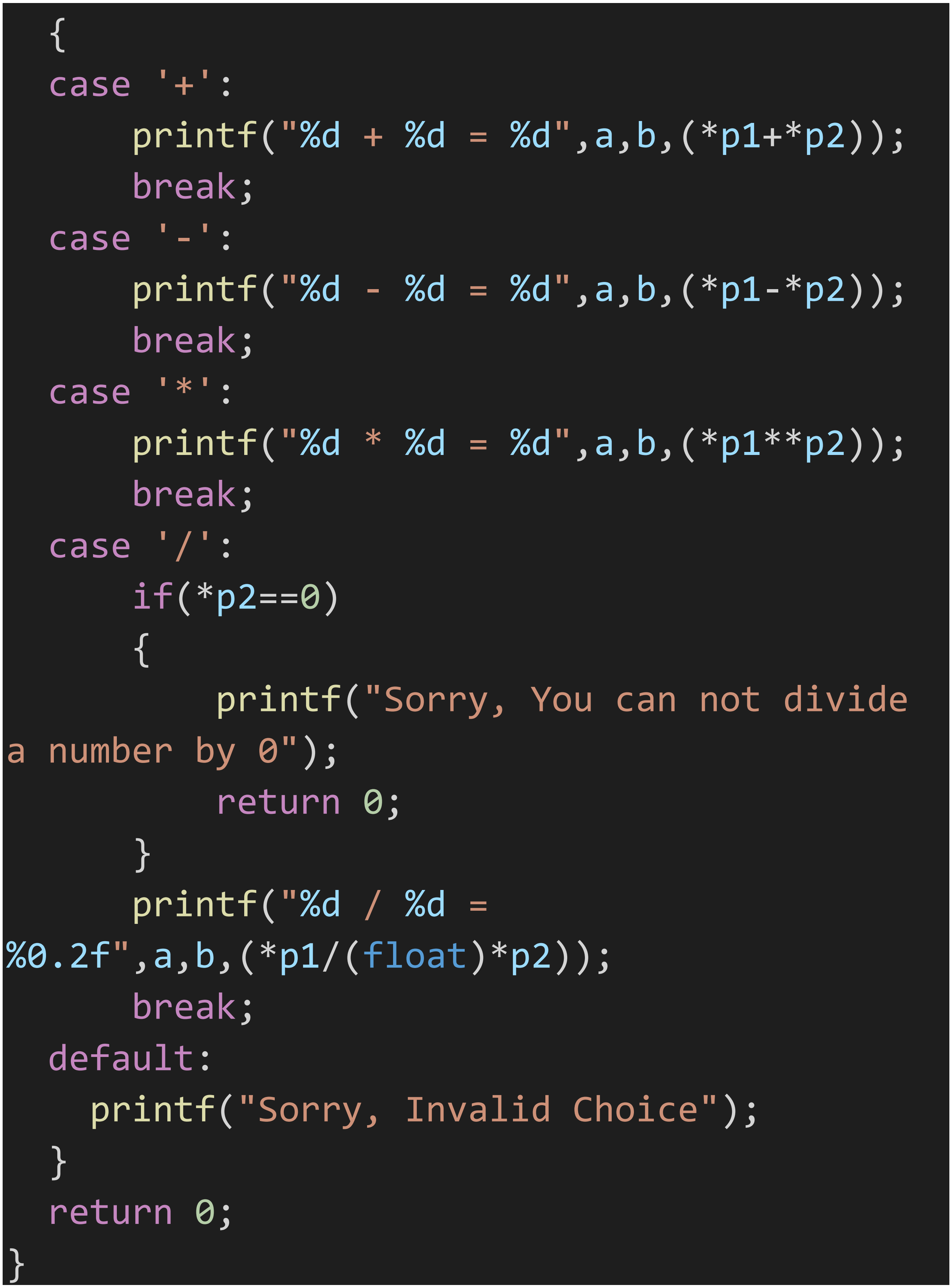
Practical :- 2

AIM :- Write a c program to create a calculator using a pointer.

Code :-

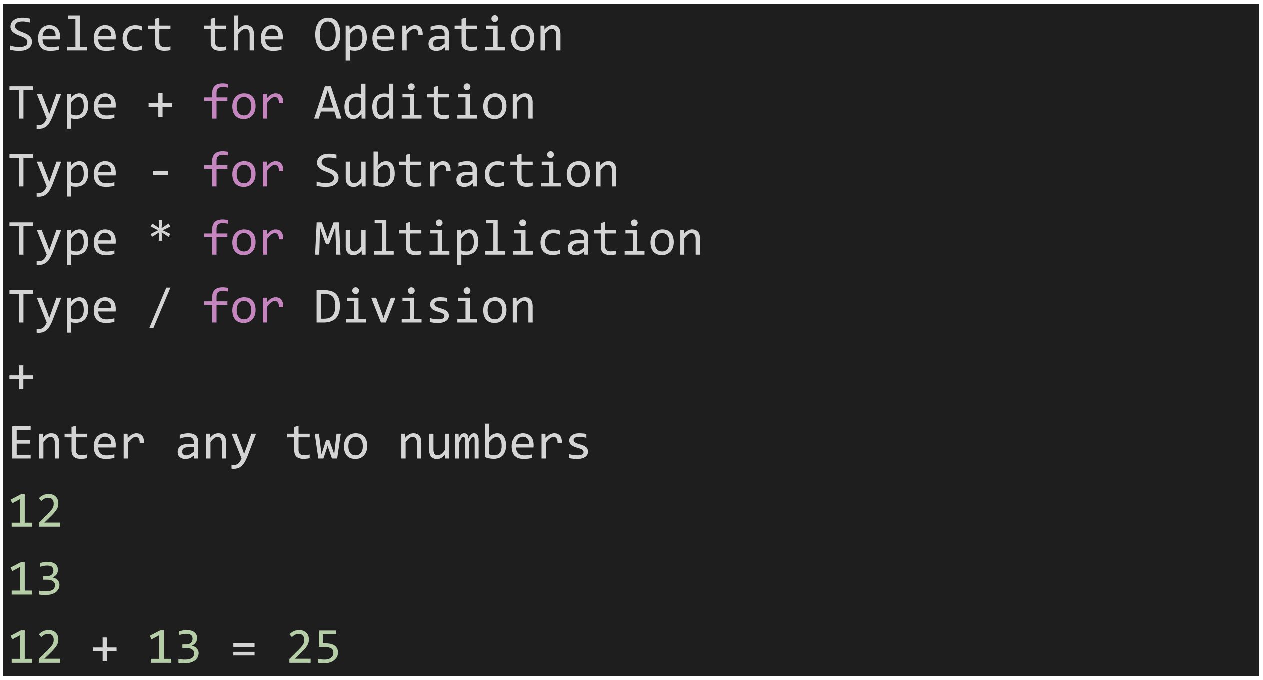


```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
  int a,b;
  int *p1,*p2;
  char ch;
  p1=&a;
  p2=&b;
  printf("Select the Operation\n");
  printf("Type + for Addition\n");
  printf("Type - for Subtraction\n");
  printf("Type * for Multiplication\n");
  printf("Type / for Division\n");
  scanf("%c",&ch);
  printf("Enter any two numbers\n");
  scanf("%d%d",&a,&b);
  switch(ch)
```



```
 {
 case '+':
   printf("%d + %d = %d",a,b,(*p1+*p2));
   break;
 case '-':
   printf("%d - %d = %d",a,b,(*p1-*p2));
   break;
 case '*':
   printf("%d * %d = %d",a,b,(*p1**p2));
   break;
 case '/':
   if(*p2==0)
   {
     printf("Sorry, You can not divide
a number by 0");
     return 0;
   }
   printf("%d / %d =
%0.2f",a,b,(*p1/(float)*p2));
   break;
 default:
  printf("Sorry, Invalid Choice");
 }
 return 0;
}
```

Output :-

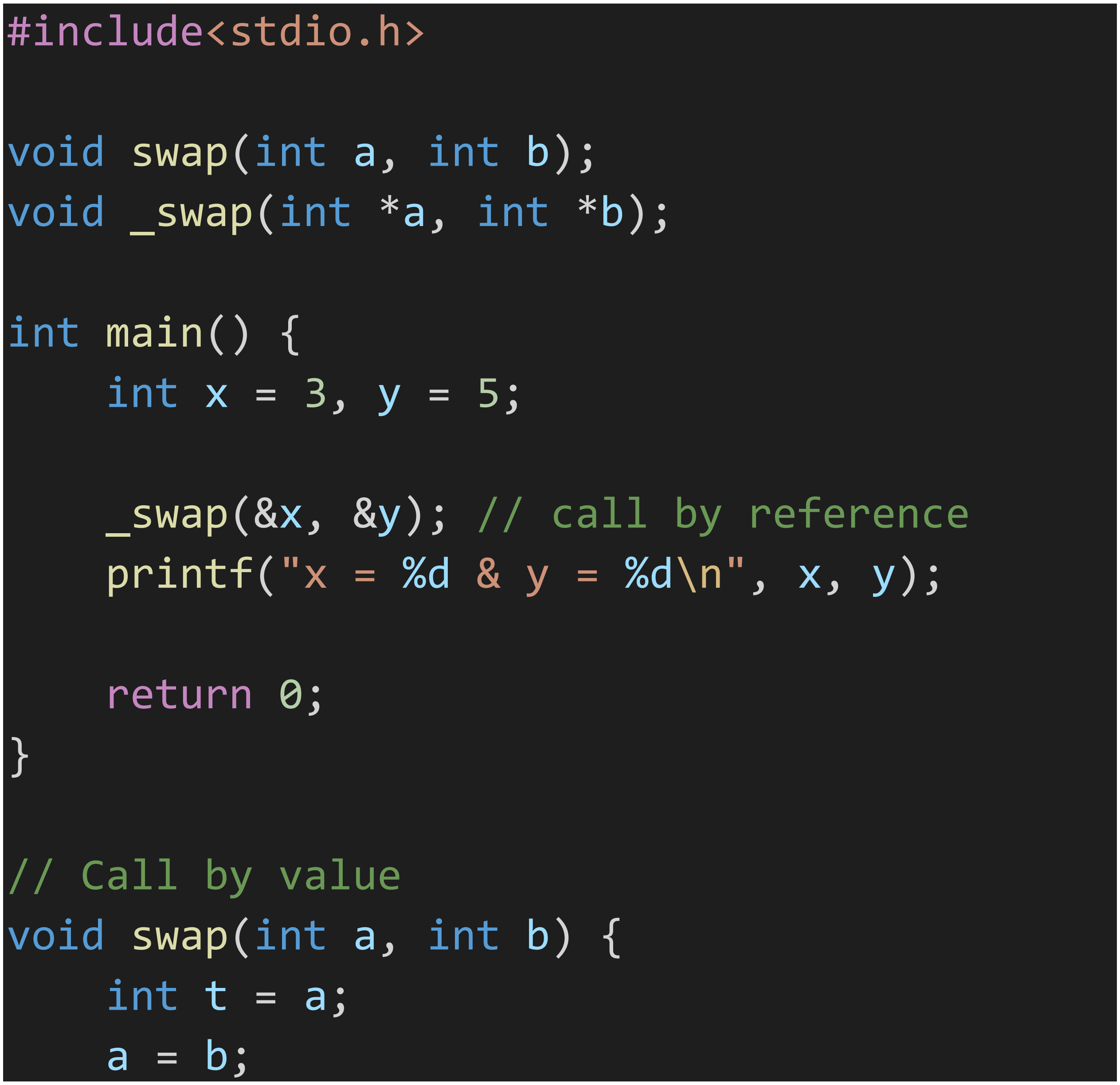


```
Select the Operation
Type + for Addition
Type - for Subtraction
Type * for Multiplication
Type / for Division
+
Enter any two numbers
12
13
12 + 13 = 25
```

Practical :- 3

AIM :- Write a c program to swap the two values using call by value and call by reference.

Code :-



```c
#include<stdio.h>

void swap(int a, int b);
void _swap(int *a, int *b);

int main() {
    int x = 3, y = 5;

    _swap(&x, &y); // call by reference
    printf("x = %d & y = %d\n", x, y);

    return 0;
}

// Call by value
void swap(int a, int b) {
    int t = a;
    a = b;
    b = t;
    printf("a = %d & b = %d\n", a, b);
}


// Call by reference
void _swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
```

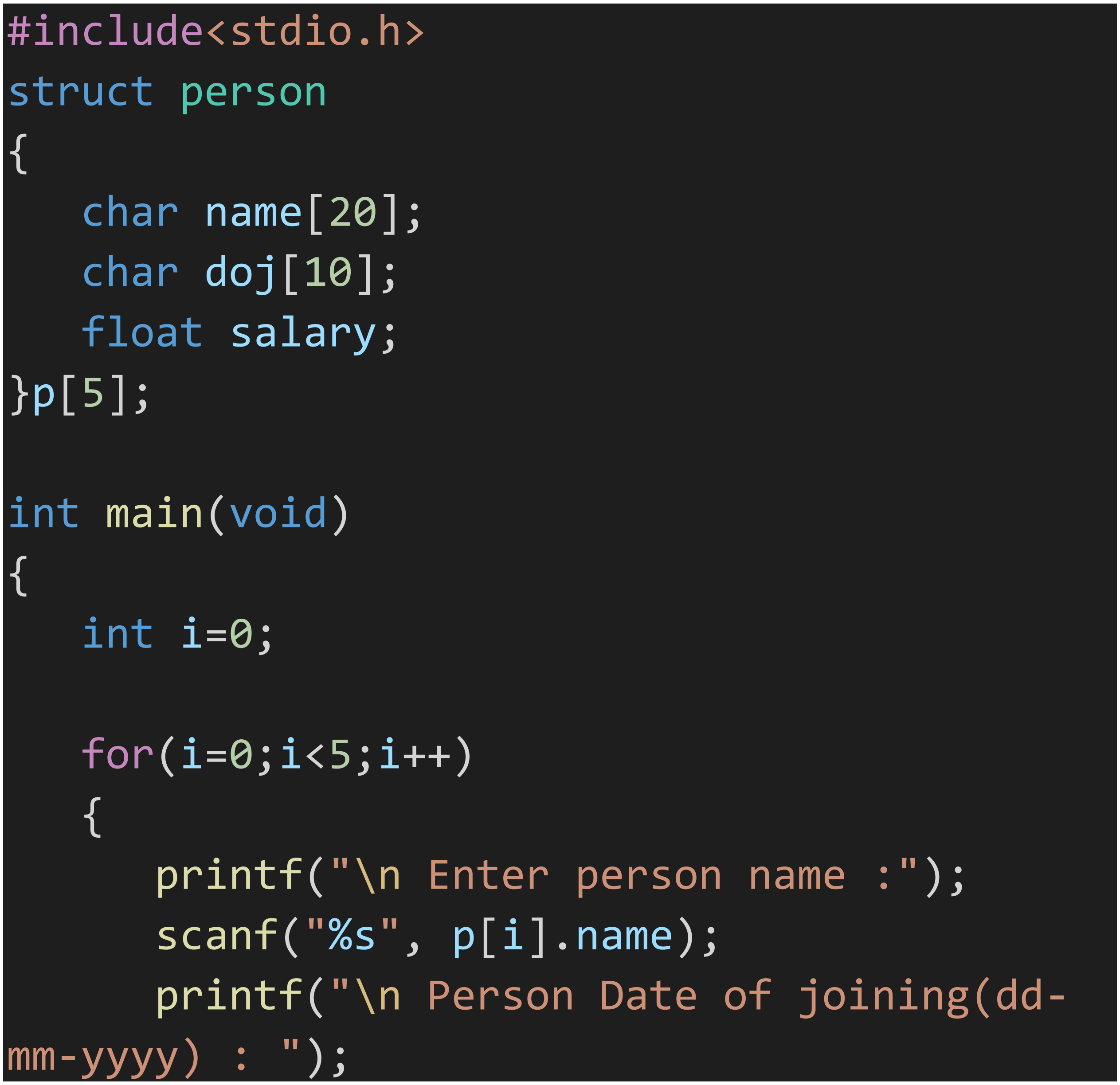
Output :-

```
Value of x and y before swap :
X = 3 and Y = 5
Value of x and y after swap :
x = 5 & y = 3
```

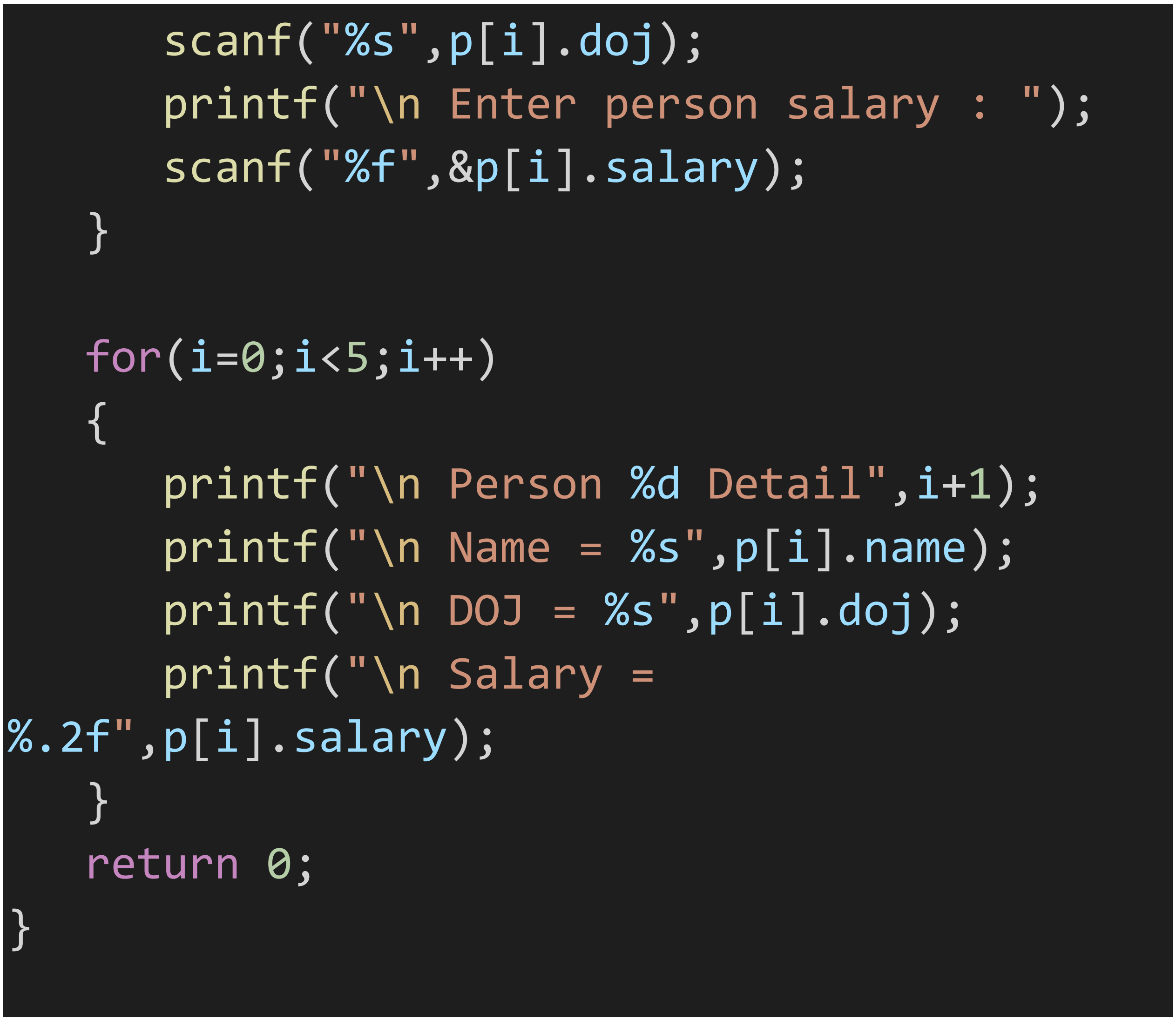
Practical :- 4

AIM :- Define a structure type struct personal that would contain person name, Date of birth and age using this structure to read this information of 4 people and display the same.

Code :-

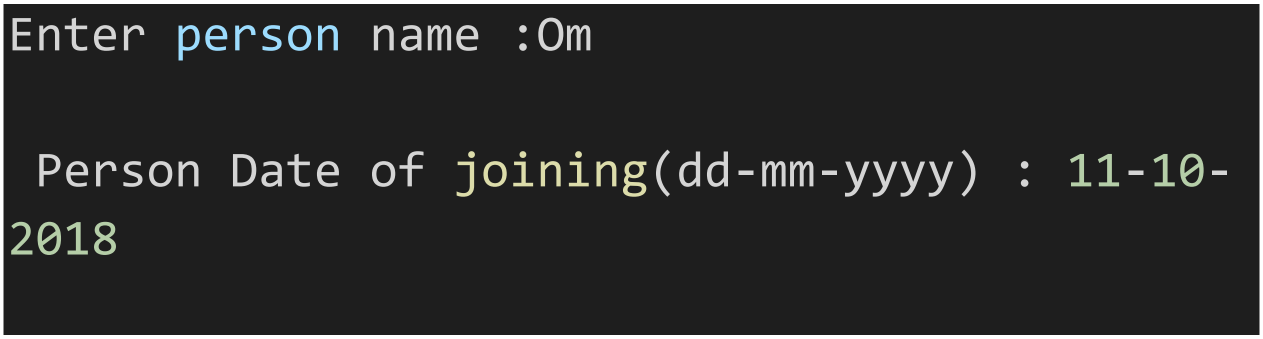


```
#include<stdio.h>
struct person
{
   char name[20];
   char doj[10];
   float salary;
}p[5];

int main(void)
{
   int i=0;

   for(i=0;i<5;i++)
   {
      printf("\n Enter person name :");
      scanf("%s", p[i].name);
      printf("\n Person Date of joining(dd-mm-yyyy) : ");
```



```c
        scanf("%s",p[i].doj);
        printf("\n Enter person salary : ");
        scanf("%f",&p[i].salary);
    }

    for(i=0;i<5;i++)
    {
        printf("\n Person %d Detail",i+1);
        printf("\n Name = %s",p[i].name);
        printf("\n DOJ = %s",p[i].doj);
        printf("\n Salary =
%.2f",p[i].salary);
    }
    return 0;
}
```

<u>Output :-</u>



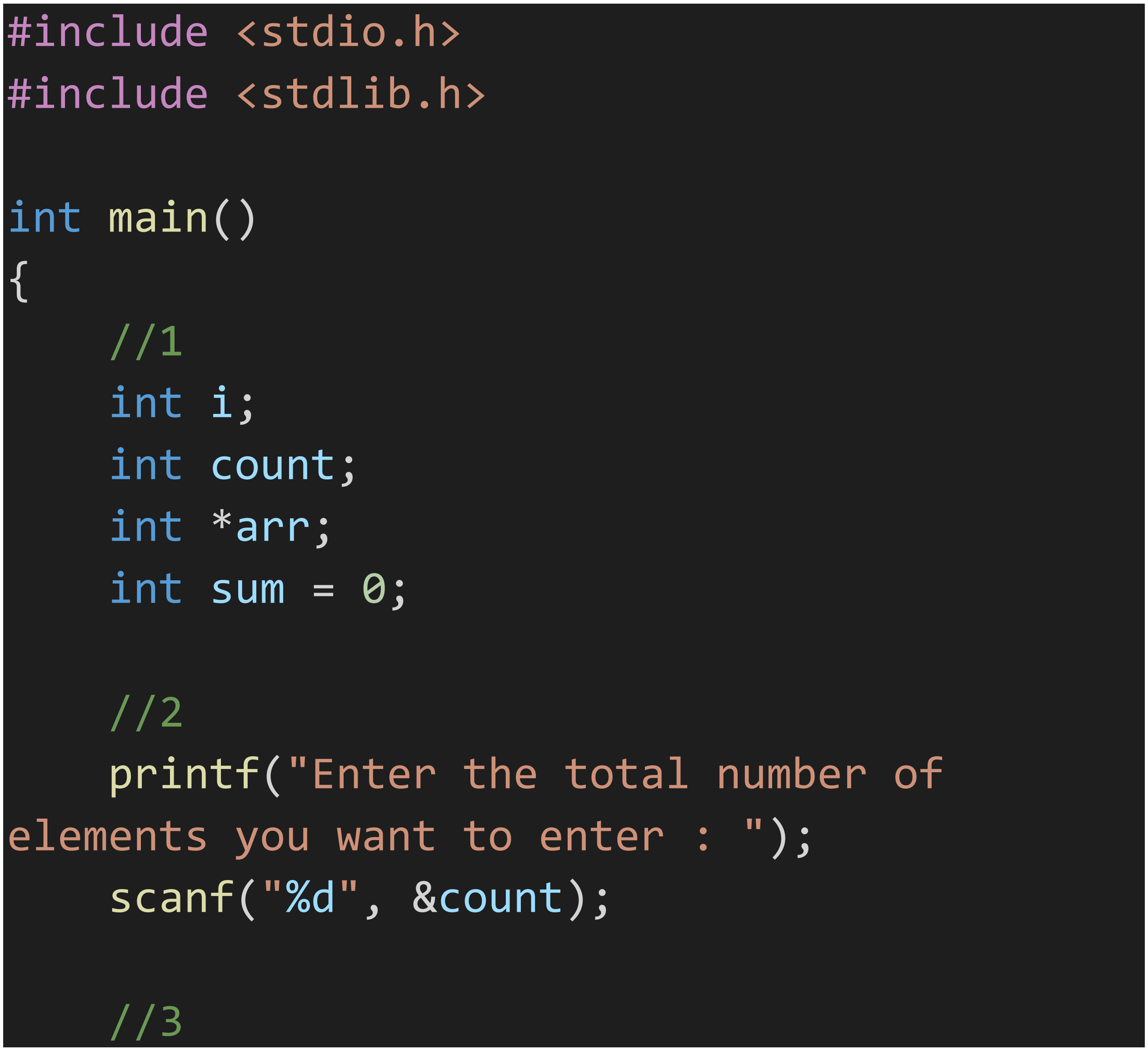
```
Enter person name :Om

 Person Date of joining(dd-mm-yyyy) : 11-10-
2018
```

```
 Enter person salary : 20000

 Enter person name :Dhruv

 Person Date of joining(dd-mm-yyyy) : 21-06-2018

 Enter person salary : 25000

 Enter person name :Ritesh

 Person Date of joining(dd-mm-yyyy) : 04-04-2019

 Enter person salary : 28000

 Enter person name :Kriti

 Person Date of joining(dd-mm-yyyy) : 26-05-2019

 Enter person salary : 30000

 Enter person name :Denisa
 Person Date of joining(dd-mm-yyyy) : 06-03-1900
```

Practical :- 5

AIM :- Write a C program to calculate the sum of n numbers entered by the user using dynamic memory allocation.

Code :-

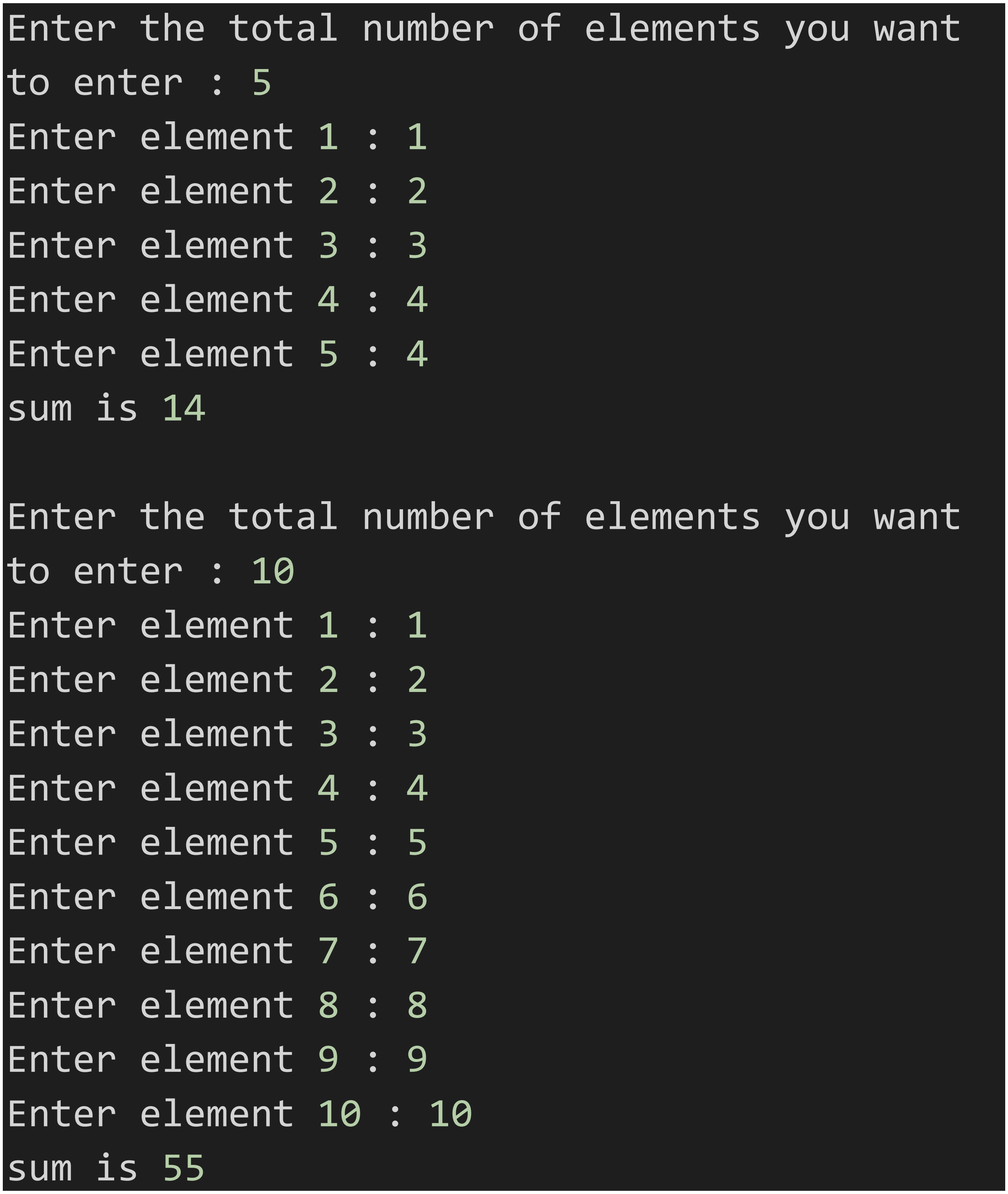


```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    //1
    int i;
    int count;
    int *arr;
    int sum = 0;

    //2
    printf("Enter the total number of 
elements you want to enter : ");
    scanf("%d", &count);

    //3
```

Output :-

```
    arr = (int *)malloc(count * 
sizeof(int));

    //4
    for (i = 0; i < count; i++)
    {
        //5
        printf("Enter element %d : ", (i + 
1));
        scanf("%d", arr + i);

        //6
        sum += *(arr + i);
    }

    //7
    printf("sum is %d \n", sum);

    //8
    free(arr);
    return 0;
}
```

Output :-

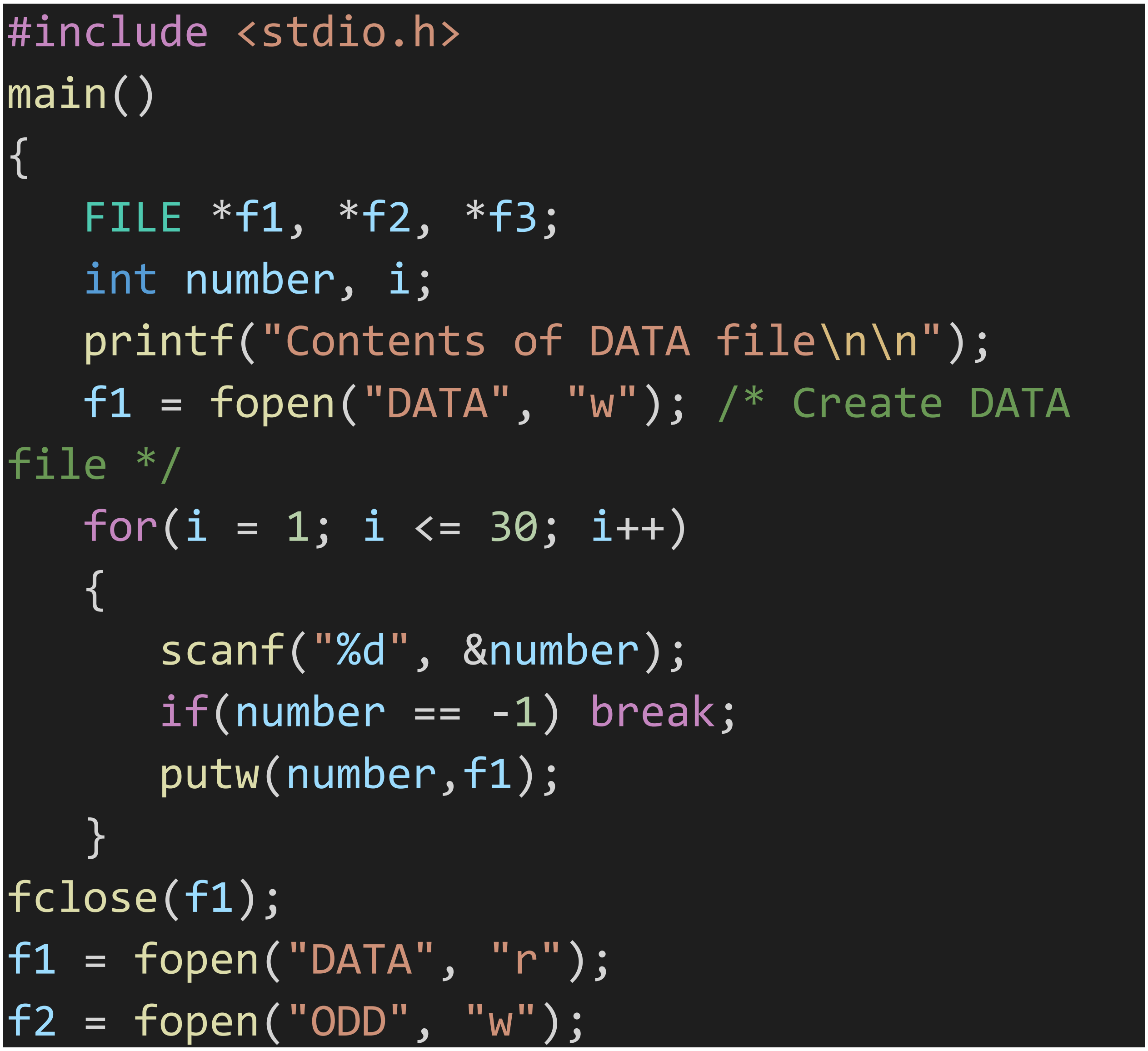


```
Enter the total number of elements you want
to enter : 5
Enter element 1 : 1
Enter element 2 : 2
Enter element 3 : 3
Enter element 4 : 4
Enter element 5 : 4
sum is 14

Enter the total number of elements you want
to enter : 10
Enter element 1 : 1
Enter element 2 : 2
Enter element 3 : 3
Enter element 4 : 4
Enter element 5 : 5
Enter element 6 : 6
Enter element 7 : 7
Enter element 8 : 8
Enter element 9 : 9
Enter element 10 : 10
sum is 55
```

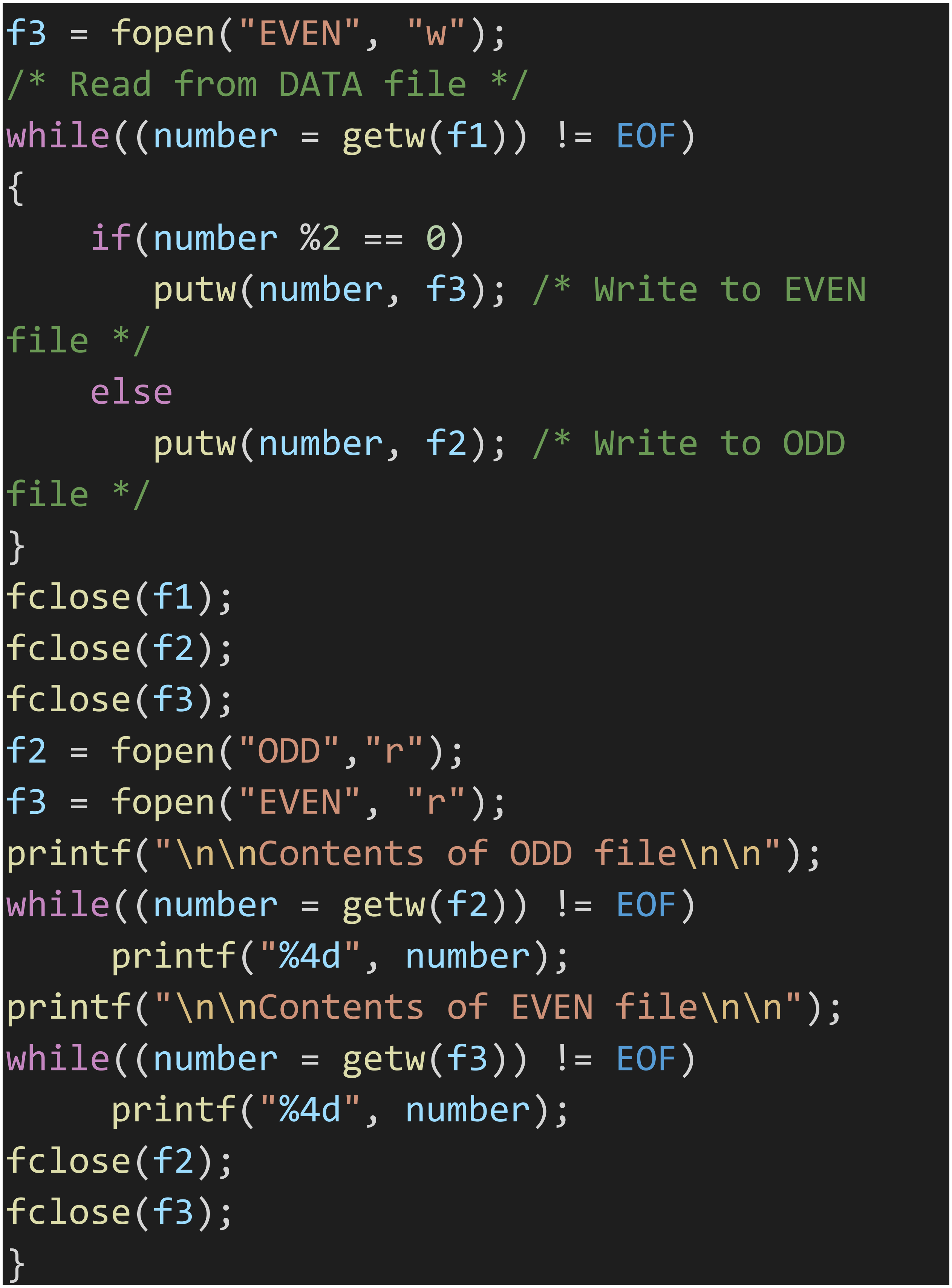
Practical :- 6

AIM :- A file named “New” contains a series of integer numbers. Write a c program to read all numbers from a file and then copy all odd numbers into a file named “odd” and write all even numbers into a file named “even”. Then display the values of files odd and even on the screen.

Code :-

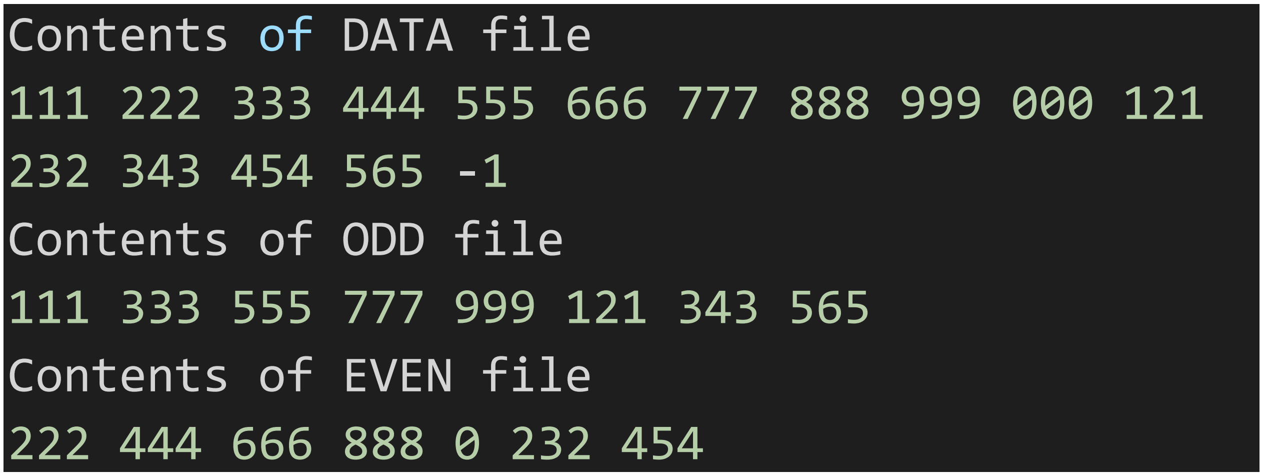


```
#include <stdio.h>
main()
{
   FILE *f1, *f2, *f3;
   int number, i;
   printf("Contents of DATA file\n\n");
   f1 = fopen("DATA", "w"); /* Create DATA
file */
   for(i = 1; i <= 30; i++)
   {
      scanf("%d", &number);
      if(number == -1) break;
      putw(number,f1);
   }
fclose(f1);
f1 = fopen("DATA", "r");
f2 = fopen("ODD", "w");
```



```
f3 = fopen("EVEN", "w");
/* Read from DATA file */
while((number = getw(f1)) != EOF)
{
    if(number %2 == 0)
       putw(number, f3); /* Write to EVEN
file */
    else
       putw(number, f2); /* Write to ODD
file */
}
fclose(f1);
fclose(f2);
fclose(f3);
f2 = fopen("ODD","r");
f3 = fopen("EVEN", "r");
printf("\n\nContents of ODD file\n\n");
while((number = getw(f2)) != EOF)
     printf("%4d", number);
printf("\n\nContents of EVEN file\n\n");
while((number = getw(f3)) != EOF)
     printf("%4d", number);
fclose(f2);
fclose(f3);
}
```

Output :-

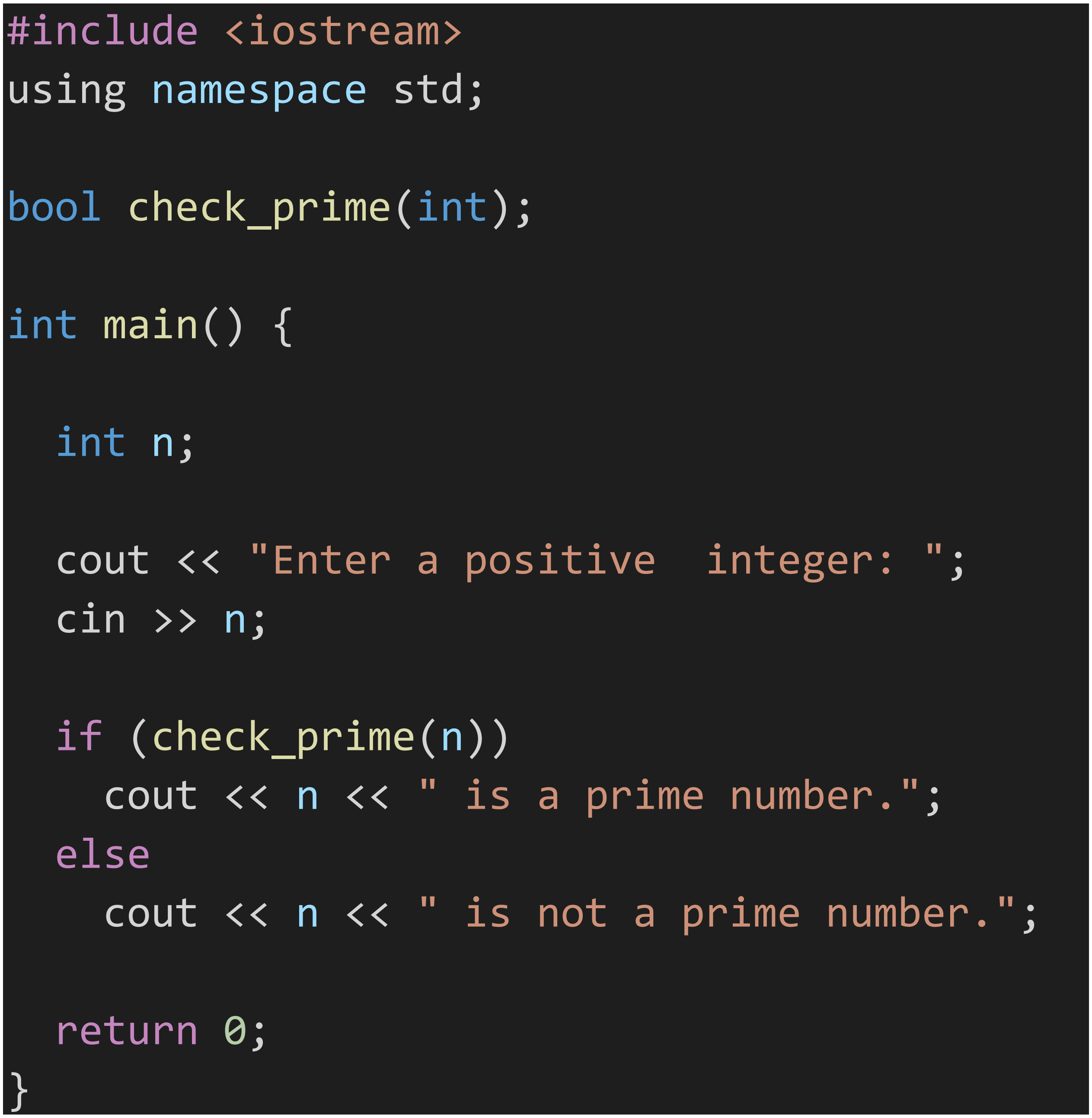


```
Contents of DATA file
111 222 333 444 555 666 777 888 999 000 121
232 343 454 565 -1
Contents of ODD file
111 333 555 777 999 121 343 565
Contents of EVEN file
222 444 666 888 0 232 454
```

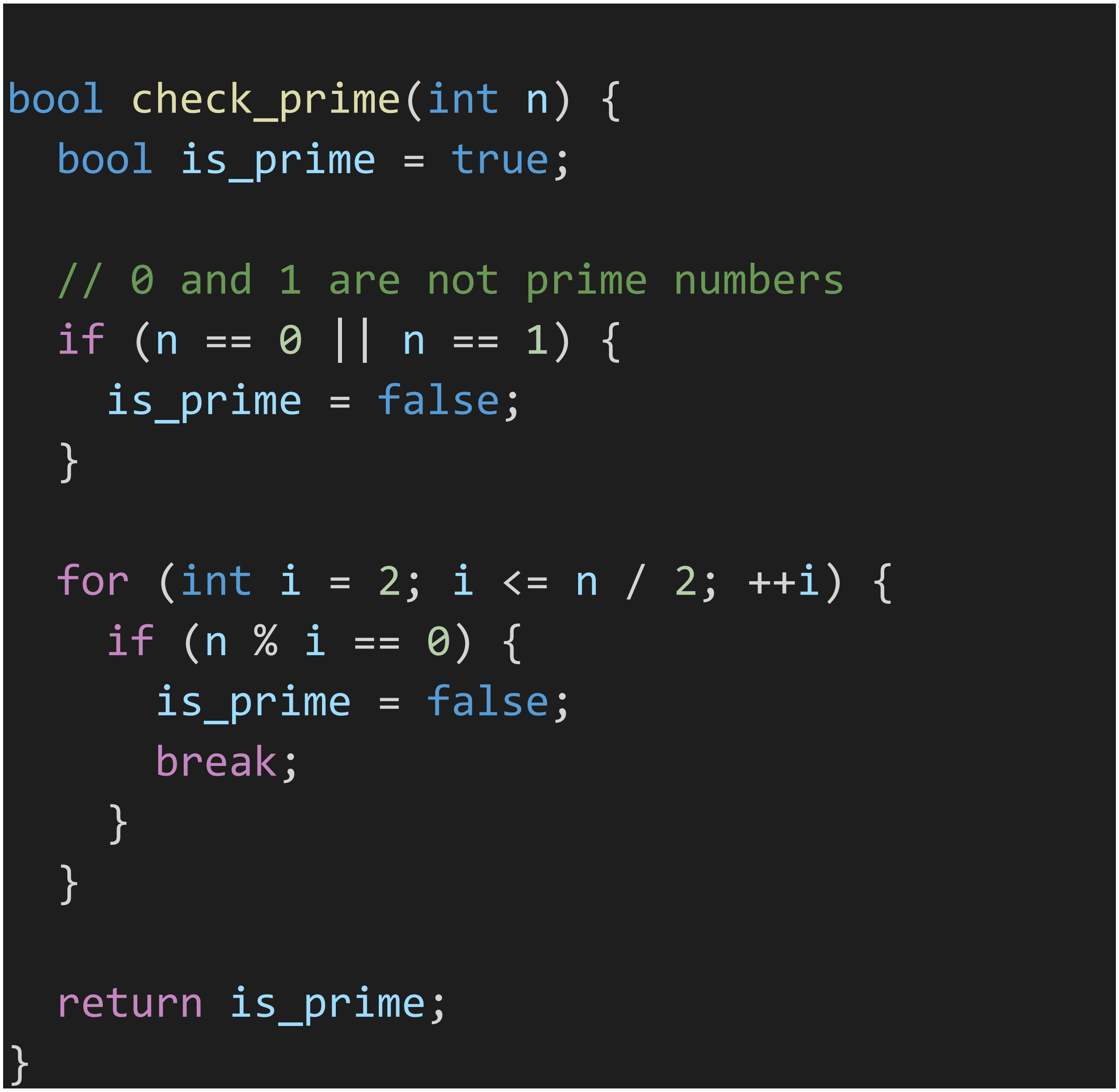
Practical :- 7

AIM :- Write a C++ program to Check if the number is prime or not using a function.

Code :-

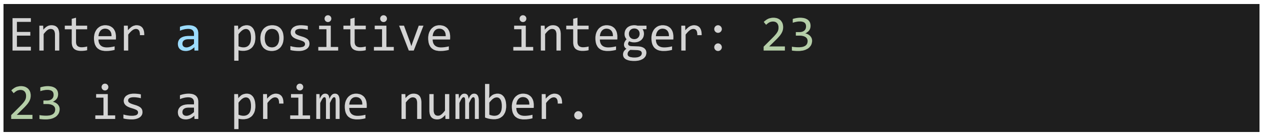


```
#include <iostream>
using namespace std;

bool check_prime(int);

int main() {

  int n;

  cout << "Enter a positive  integer: ";
  cin >> n;

  if (check_prime(n))
    cout << n << " is a prime number.";
  else
    cout << n << " is not a prime number.";

  return 0;
}
```



```cpp
bool check_prime(int n) {
  bool is_prime = true;

  // 0 and 1 are not prime numbers
  if (n == 0 || n == 1) {
    is_prime = false;
  }

  for (int i = 2; i <= n / 2; ++i) {
    if (n % i == 0) {
      is_prime = false;
      break;
    }
  }

  return is_prime;
}
```

<u>Output :-</u>

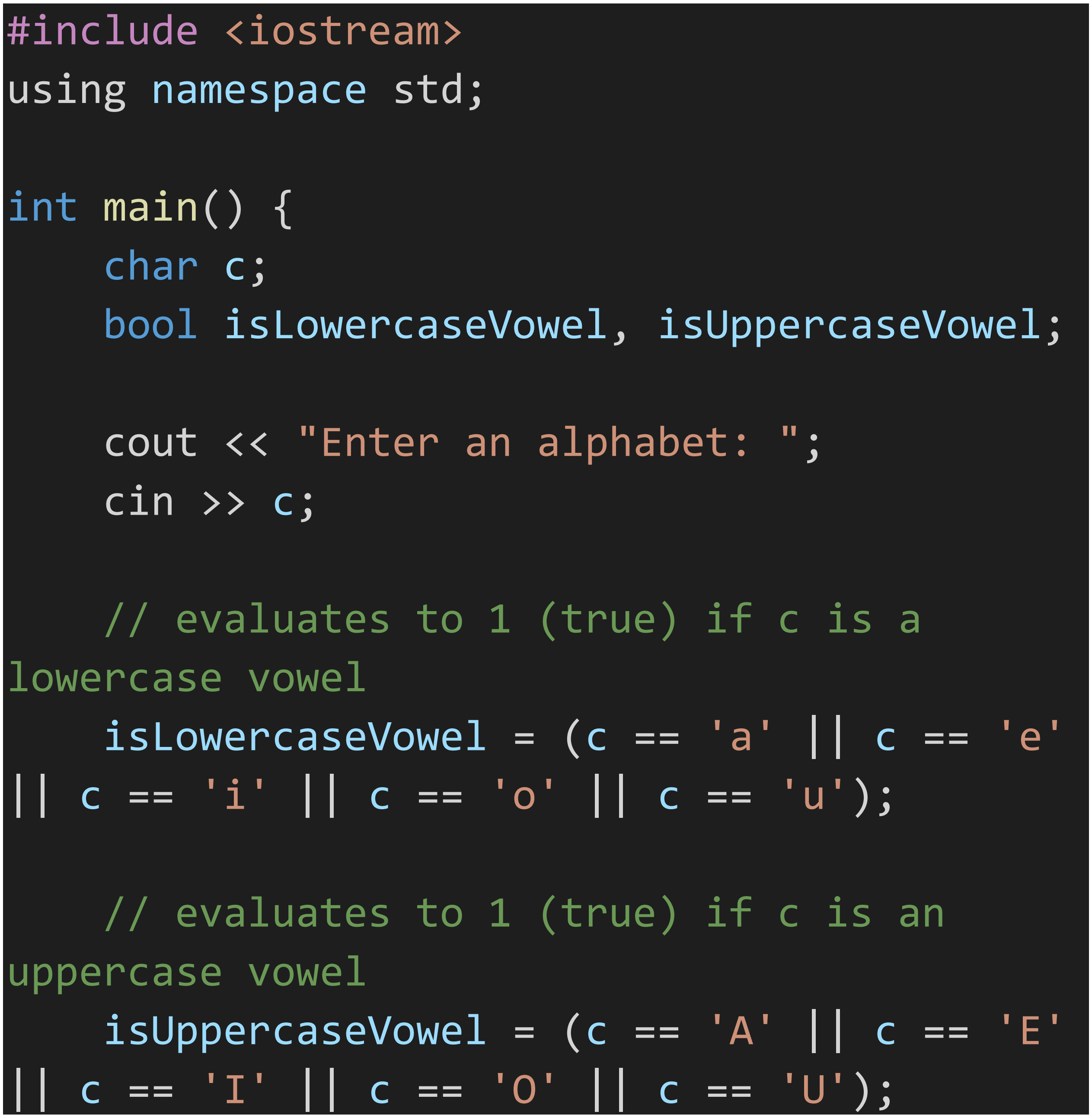


```
Enter a positive  integer: 23
23 is a prime number.
```

Practical :- 8

AIM :- Write a C++ program that prompts the user to enter a letter and check whether a letter is a vowel or constant.

Code :-



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

int main() {
    char c;
    bool isLowercaseVowel, isUppercaseVowel;

    cout << "Enter an alphabet: ";
    cin >> c;

    // evaluates to 1 (true) if c is a lowercase vowel
    isLowercaseVowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

    // evaluates to 1 (true) if c is an uppercase vowel
    isUppercaseVowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
    // show error message if c is not an
alphabet
    if (!isalpha(c))
      printf("Error! Non-alphabetic
character.");
    else if (isLowercaseVowel ||
isUppercaseVowel)
        cout << c << " is a vowel.";
    else
        cout << c << " is a consonant.";

    return 0;
}
```

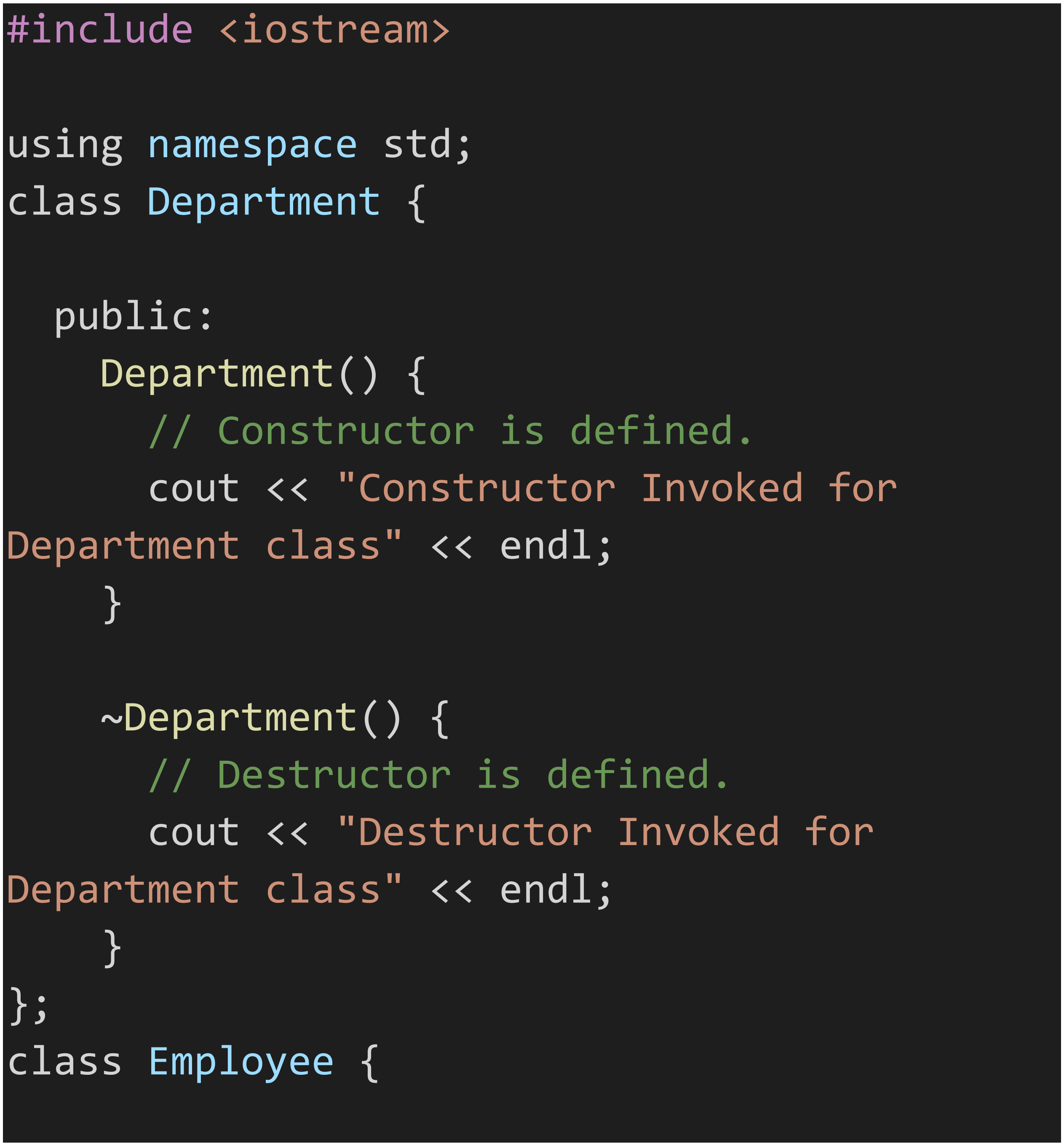
Output :-

```
Enter an alphabet: u
u is a vowel.
```

Practical :- 9

AIM :- Write a C++ program to demonstrate the concept of constructor and destructor.

Code :-



```
#include <iostream>

using namespace std;
class Department {

  public:
    Department() {
      // Constructor is defined.
      cout << "Constructor Invoked for
Department class" << endl;
    }

    ~Department() {
      // Destructor is defined.
      cout << "Destructor Invoked for
Department class" << endl;
    }
};
class Employee {
  public:
    Employee() {
      // Constructor is defined.
      cout << "Constructor Invoked for Employee class" << endl;
    }

    ~Employee() {
      // Destructor is defined.
      cout << "Destructor Invoked for Employee class" << endl;
    }
};
int main(void) {
  // Creating an object of Department.
  Department d1;
  // Creating an object of Employee.
  Employee e2;
  return 0;
}
```

Output :-

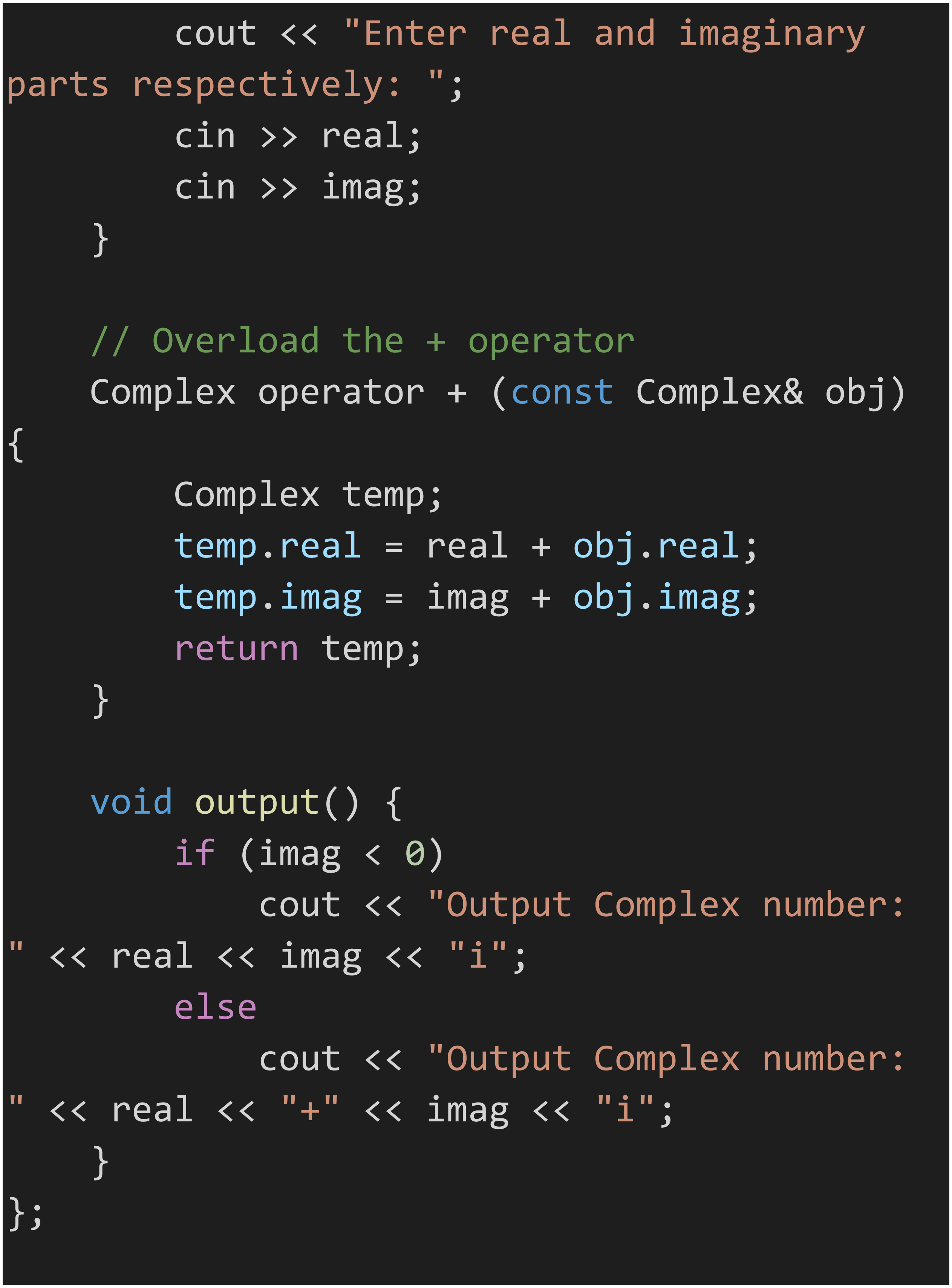
```
Constructor Invoked for Department class
Constructor Invoked for Employee class
Destructor Invoked for Employee class
Destructor Invoked for Department class
```

Practical :- 11

AIM :- Write a C++ program to overload binary + operator.

Code :-

```
// C++ program to overload the binary operator +
// This program adds two complex numbers

#include <iostream>
using namespace std;

class Complex {
   private:
    float real;
    float imag;

   public:
    // Constructor to initialize real and imag to 0
    Complex() : real(0), imag(0) {}

    void input() {
```



```
        cout << "Enter real and imaginary 
parts respectively: ";
        cin >> real;
        cin >> imag;
    }

    // Overload the + operator
    Complex operator + (const Complex& obj) 
{
        Complex temp;
        temp.real = real + obj.real;
        temp.imag = imag + obj.imag;
        return temp;
    }

    void output() {
        if (imag < 0)
            cout << "Output Complex number: 
" << real << imag << "i";
        else
            cout << "Output Complex number: 
" << real << "+" << imag << "i";
    }
};
```

```
int main() {
    Complex complex1, complex2, result;

    cout << "Enter first complex number:\n";
    complex1.input();

    cout << "Enter second complex number:\n";
    complex2.input();

   // complex1 calls the operator function
   // complex2 is passed as an argument to the function
    result = complex1 + complex2;
    result.output();

    return 0;
}
```

Output :-

```
Enter first complex number:
Enter real and imaginary parts respectively:
9 5
Enter second complex number:
```

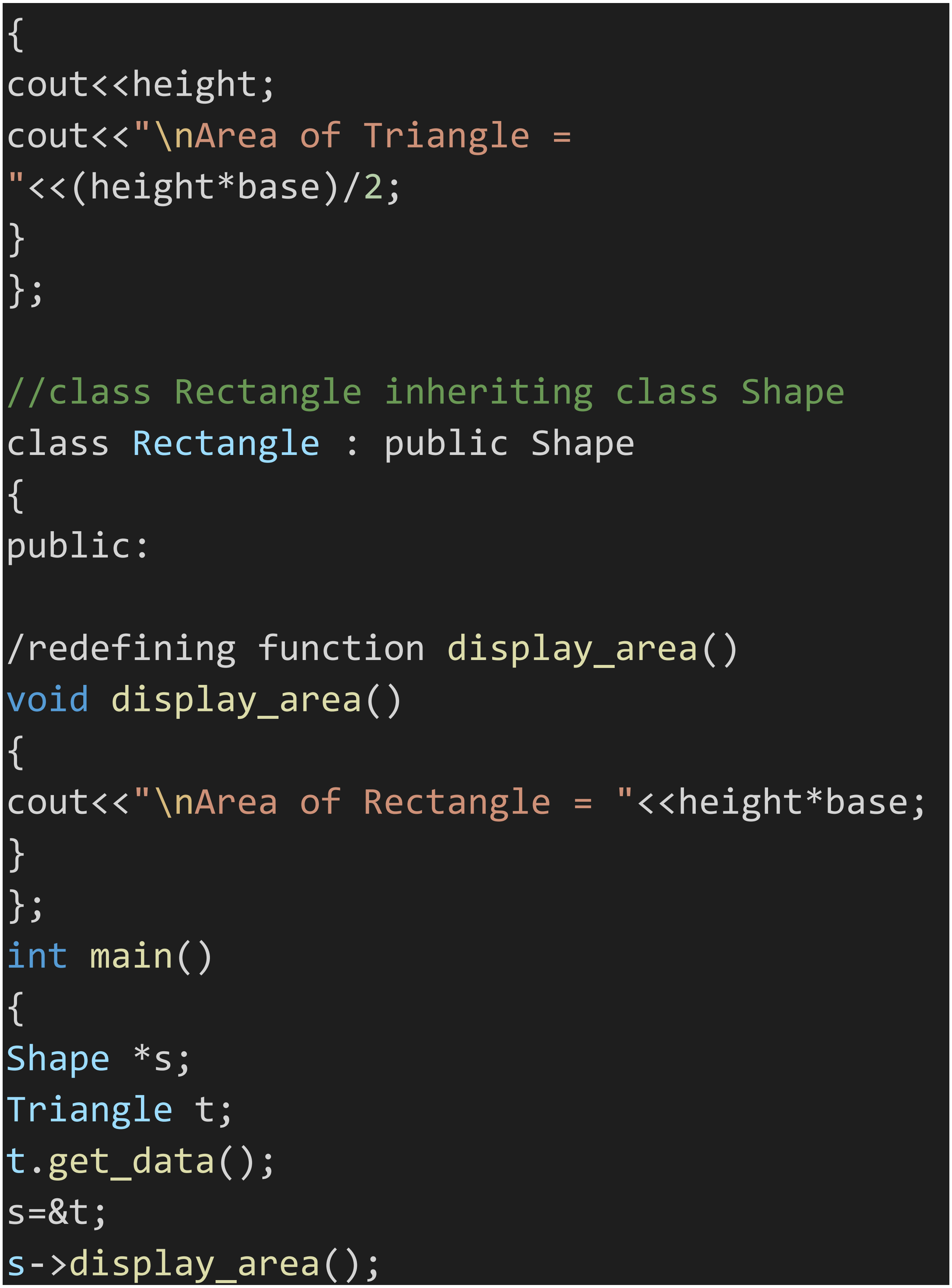
```
Enter real and imaginary parts respectively:
7 6
Output Complex number: 16+11i
```

Practical :- 12

AIM :- Create a base class called 'SHAPE' having two data members of type double, member function get_data( ) to initialize base class data members, pure virtual member function display_area( ) to compute and display the area of the geometrical object. Derive two specific classes 'TRIANGLE' and 'RECTANGLE' from the base class. Using these three classes design a program that will accept dimension of a triangle / rectangle interactively and display the area.

Code :-

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>

//class shape
class Shape
{
public:
double height,base;

//constructor to assign initial values to height and base
Shape()
{
height=0;
base=0;
}

//get_data() function to get values of
height and base
void get_data()
{
cout<<"\nEnter height and base to compute
are :";
cin>>height>>base;
}

//declaration of virtual function
display_area()
virtual void display_area()
{
}
};

//class triangle inheriting class Shape
class Triangle : public Shape
{
public:

//redefining function display_area()
void display_area()
```



```
{
cout<<height;
cout<<"\nArea of Triangle =
"<<(height*base)/2;
}
};

//class Rectangle inheriting class Shape
class Rectangle : public Shape
{
public:

/redefining function display_area()
void display_area()
{
cout<<"\nArea of Rectangle = "<<height*base;
}
};
int main()
{
Shape *s;
Triangle t;
t.get_data();
s=&t;
s->display_area();
```

```
Rectangle r;
r.get_data();
s=&r;
s->display_area();
return 0;
}
```

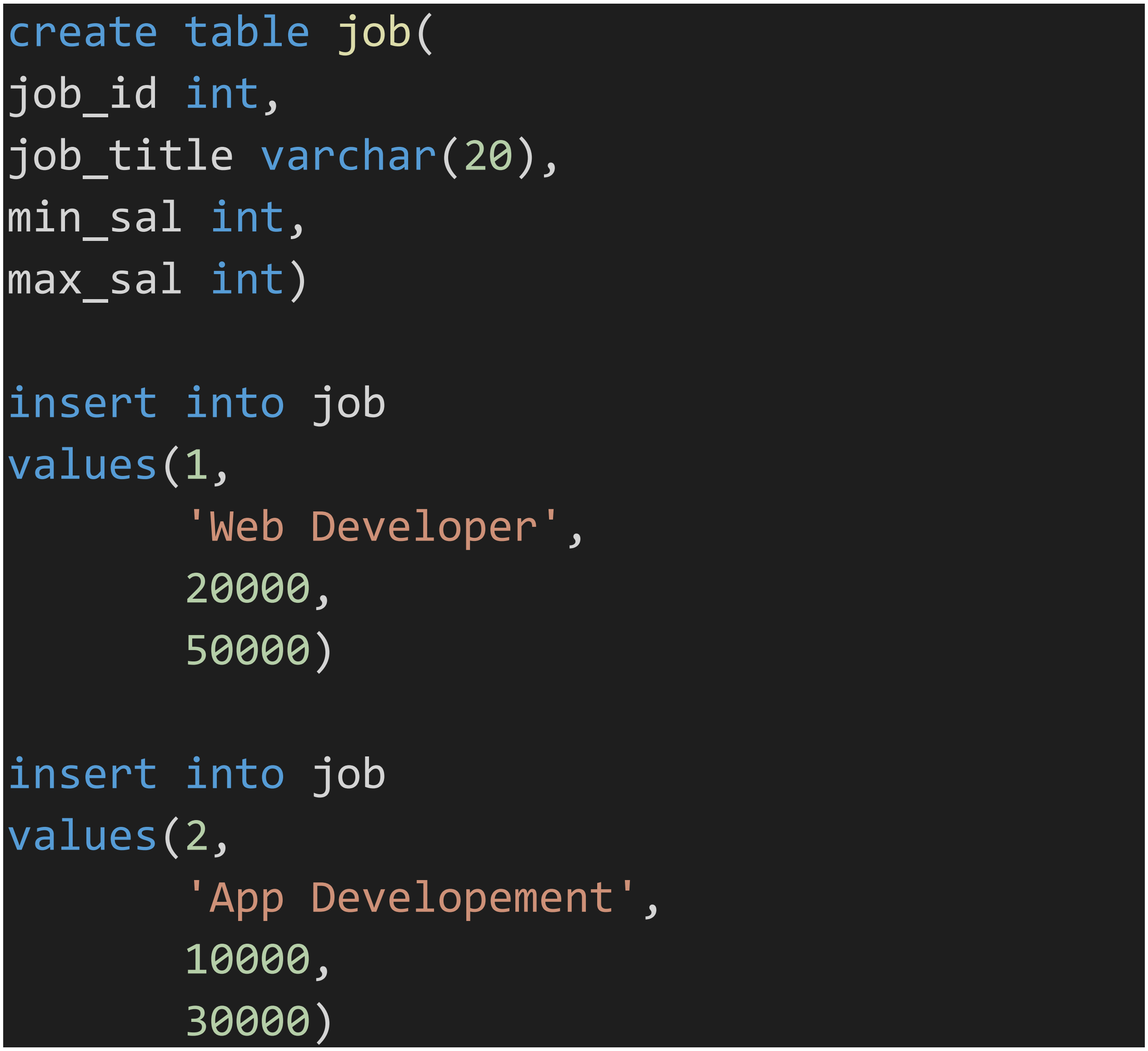
Output :-

```
Enter height and base to compute are :10
7
10
Area of Triangle = 35
Enter height and base to compute are :10
20

Area of Rectangle = 200
```

Practical :- 13

AIM :- To study DDL-create and DML-insert commands.
Create following Table :- Job (job_id, job_title, min_sal, max_sal)

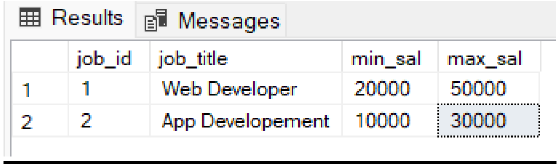
Code :-



```
create table job(
job_id int,
job_title varchar(20),
min_sal int,
max_sal int)

insert into job
values(1,
       'Web Developer',
       20000,
       50000)

insert into job
values(2,
       'App Developement',
       10000,
       30000)
```

```sql
select * from job;
```

Output :-



	job_id	job_title	min_sal	max_sal
1	1	Web Developer	20000	50000
2	2	App Developement	10000	30000

Practical :- 14

AIM-1 :- Job (job_id, job_title, min_sal, max_sal)

Code :-

```
create table Job(
COLUMN_NAME VARCHAR(15),
DATA_TYPE VARCHAR(15))

insert into Job
values('job_id',
       'Varchar(15)')

insert into Job
values('job_title',
       'Varchar(30)')

insert into Job
values('min_sal',
       'Int')

insert into Job
values('max_sal',
       'Int')

select * from Job;
```

AIM-2 :- Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no)

Code :-

```
create table Employee(
COLUMN_NAME VARCHAR(15),
DATA_TYPE VARCHAR(15))

insert into Employee
values('emp_no',
       'Int')

insert into Employee
values('emp_name',
       'Varchar(30)')

insert into Employee
values('emp_sal',
       'decimal(8,2)')
```

```
insert into Employee
values('emp_comm',
       'decimal(6,1)')

insert into Employee
values('dept_no',
       'Int')

select * from Employee;
```

AIM-3 :- deposit(a_no,cname,bname,amount,a_date)

Code :-

```
create table deposite(
COLUMN_NAME VARCHAR(15),
DATA_TYPE VARCHAR(15))

insert into deposite
values('a_no',
       'Int,identity')

insert into deposite
values('cname',
       'Varchar(50)')
```

```
insert into deposite
values('bname',
       'Varchar(30)')

insert into deposite
values('emp_comm',
       'decimal(6,1)')

insert into deposite
values('amount',
       'Decimal(4,2)')

insert into deposite
values('a_date',
       'Date')

select * from deposite;
```

AIM-4 :- borrow(loanno,cname,bname,amount)

Code :-

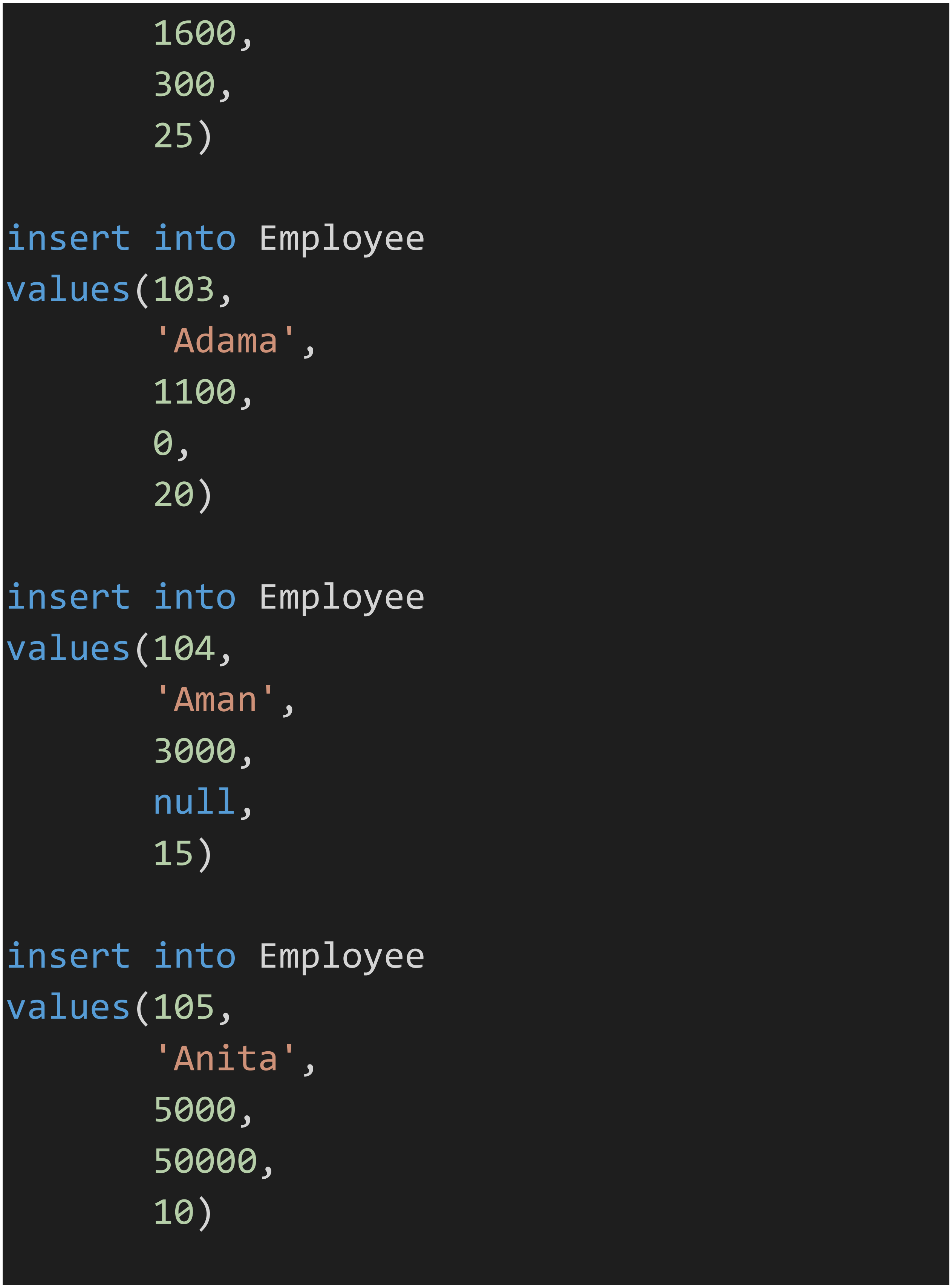
```
create table borrow(
COLUMN_NAME VARCHAR(15),
DATA_TYPE VARCHAR(15))

insert into borrow
values('loan_no',
       'Int')

insert into borrow
values('cname',
       'Varchar(25)')

insert into borrow
values('bname',
       'Varchar(20)')

insert into borrow
values('emp_comm',
       'decimal(6,1)')

insert into borrow
values('amount',
       'Decimal(6,2)')

select * from borrow;
```

Practical :- 15

AIM :- Create tables and insert sample data in tables. Write SQL queries to insert following data into tables

Code-1 :-

```
create table Employee(
emp_n int,
emp_name varchar(15),
emp_sal int,
emp_comm int,
dept_no int)

insert into Employee
values(101,
       'Smit',
       800,
       null,
       20)

insert into Employee
values(102,
       'Snehal',
```

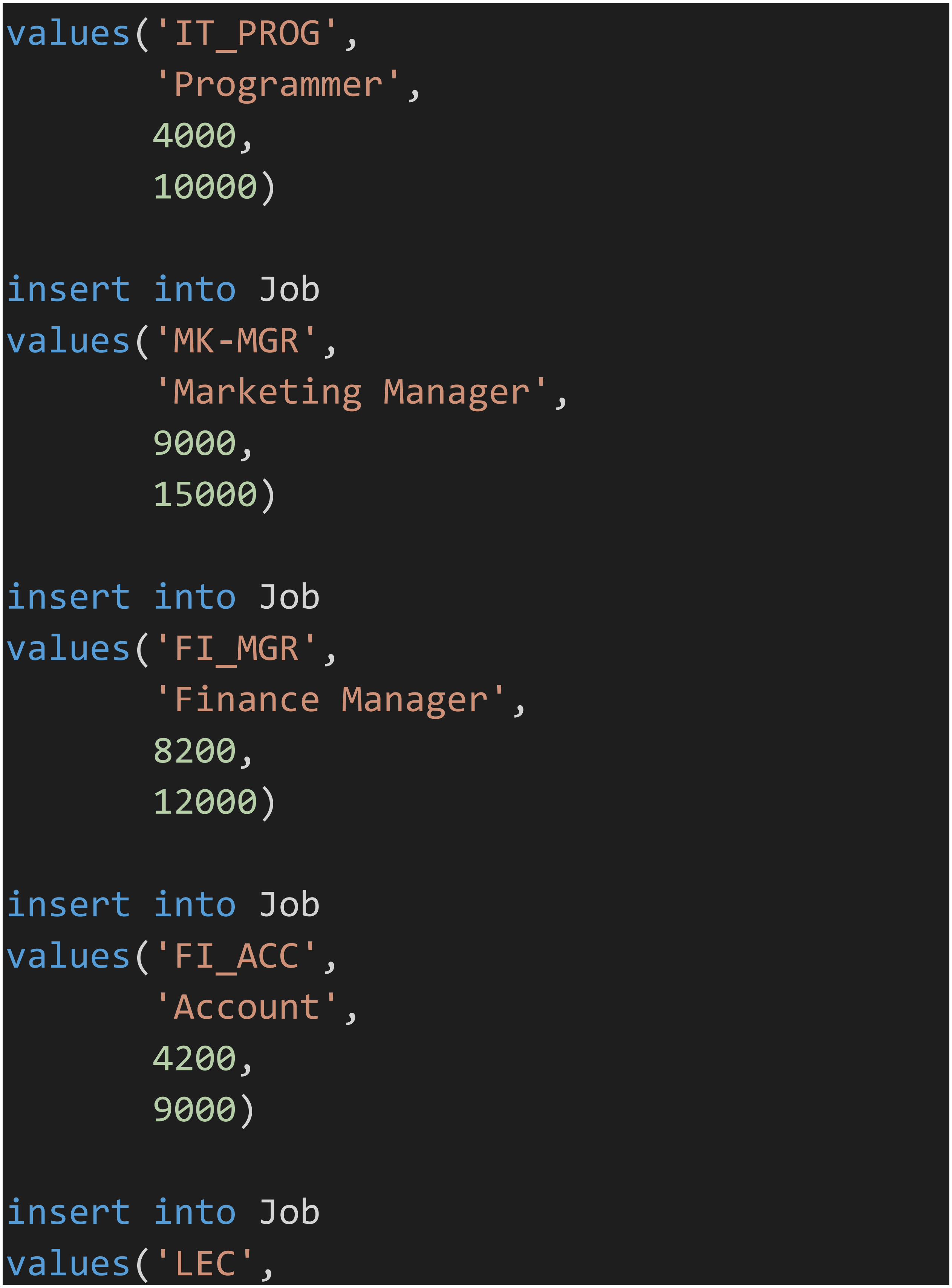


```
       1600,
       300,
       25)

insert into Employee
values(103,
       'Adama',
       1100,
       0,
       20)

insert into Employee
values(104,
       'Aman',
       3000,
       null,
       15)

insert into Employee
values(105,
       'Anita',
       5000,
       50000,
       10)
```

```
insert into Employee
values(106,
       'Sneha',
       2450,
       24500,
       10)

insert into Employee
values(107,
       'Anamika',
       2975,
       null,
       30)

select * from Employee;
```

Code-2 :-

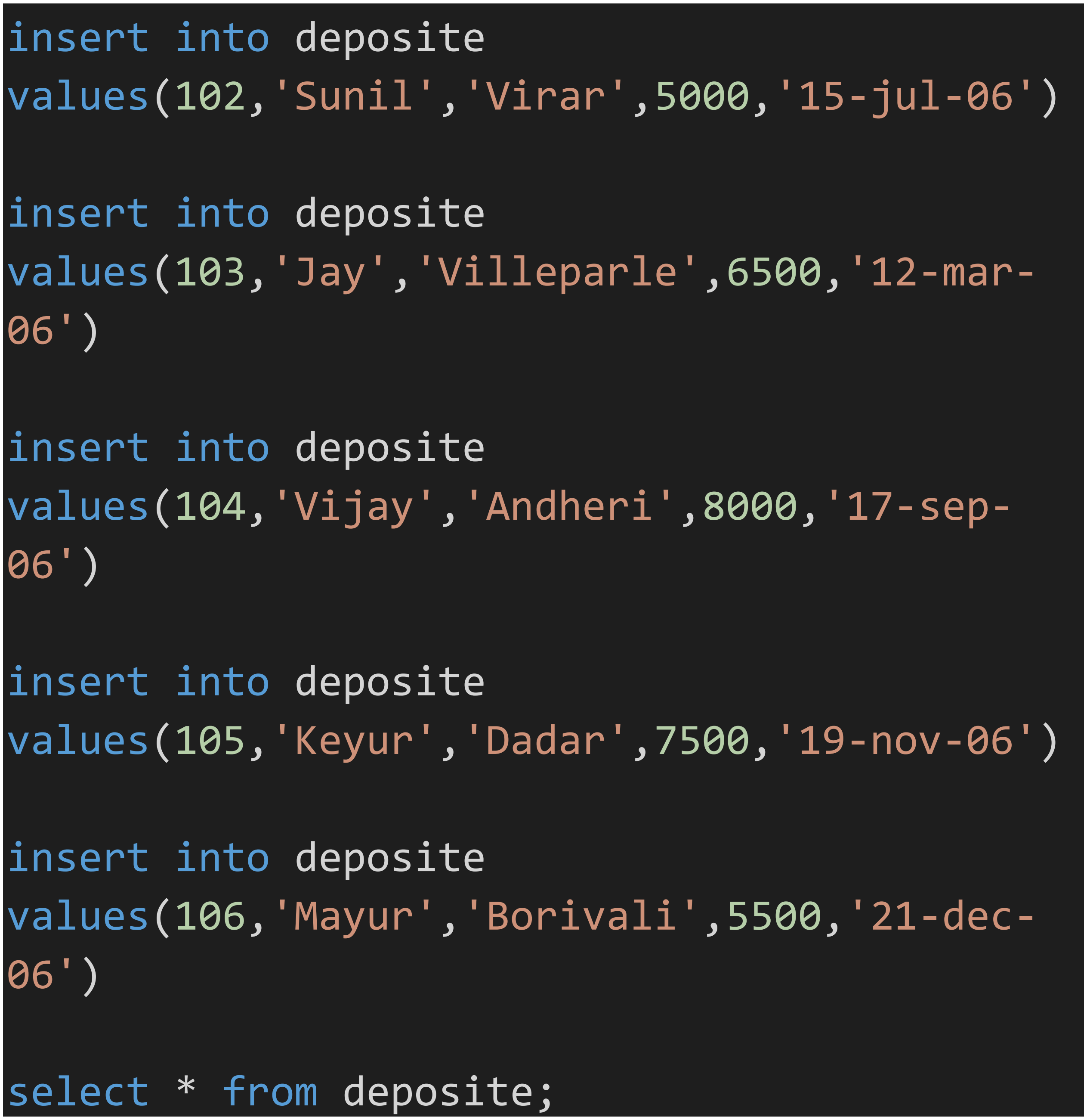
```
create table Job(
job_id varchar(20),
job_name varchar(20),
min_sal int,
max_sal int)

insert into Job
```



```
values('IT_PROG',
       'Programmer',
       4000,
       10000)

insert into Job
values('MK-MGR',
       'Marketing Manager',
       9000,
       15000)

insert into Job
values('FI_MGR',
       'Finance Manager',
       8200,
       12000)

insert into Job
values('FI_ACC',
       'Account',
       4200,
       9000)

insert into Job
values('LEC',
       'Lecture',
       6000,
       17000)

insert into Job
values('COMP_OP',
       'Computer Oparator',
       1500,
       3000)

select * from Job;
```

Code-3 :-

```
create table deposite(
A_no int,
cname varchar(10),
Bname varchar(15),
Amount int,
date date)

insert into deposite
values(101,'Anil','Andheri',7000,'01-jan-
06')
```

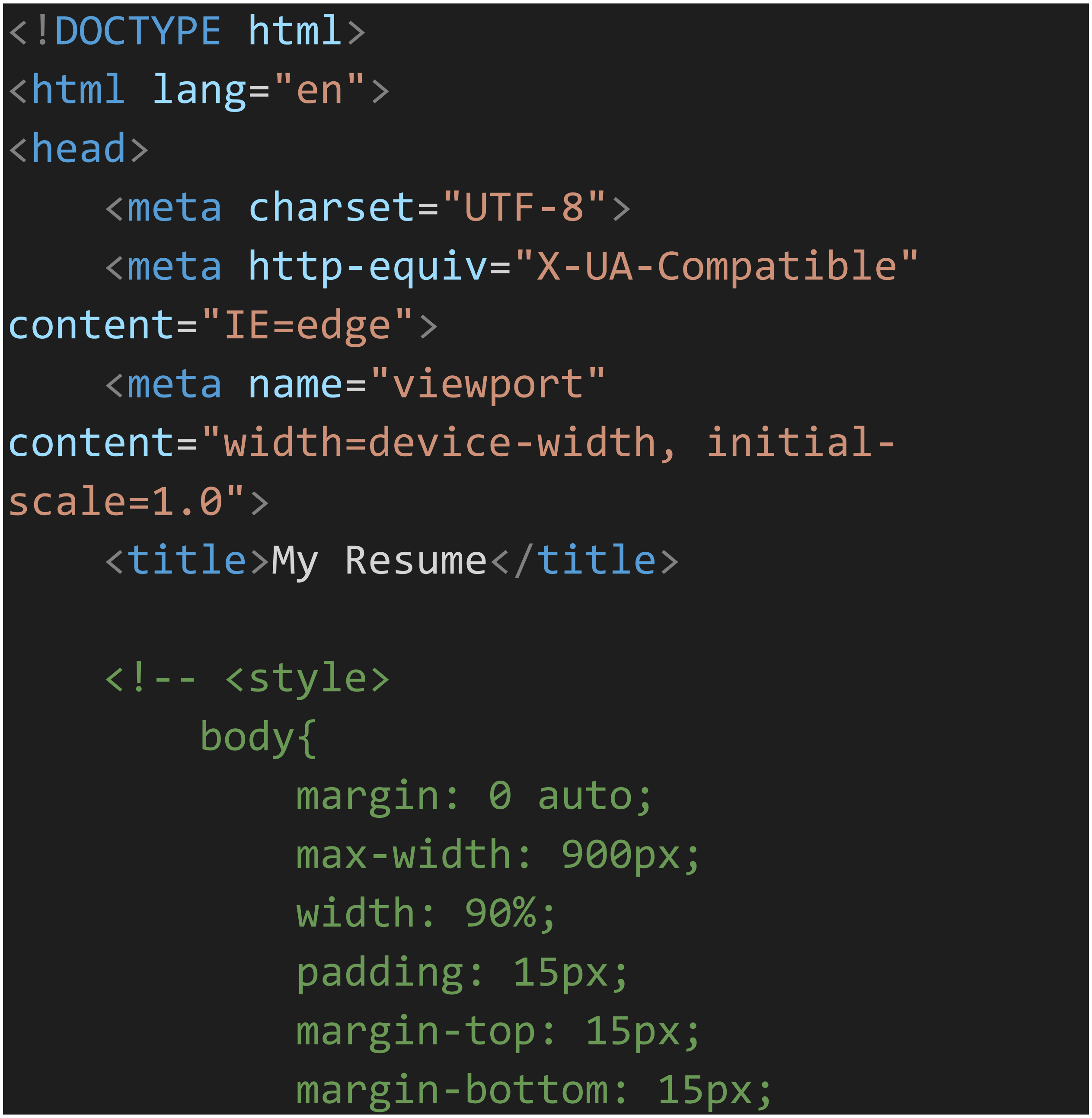


```
insert into deposite
values(102,'Sunil','Virar',5000,'15-jul-06')

insert into deposite
values(103,'Jay','Villeparle',6500,'12-mar-
06')

insert into deposite
values(104,'Vijay','Andheri',8000,'17-sep-
06')

insert into deposite
values(105,'Keyur','Dadar',7500,'19-nov-06')

insert into deposite
values(106,'Mayur','Borivali',5500,'21-dec-
06')

select * from deposite;
```

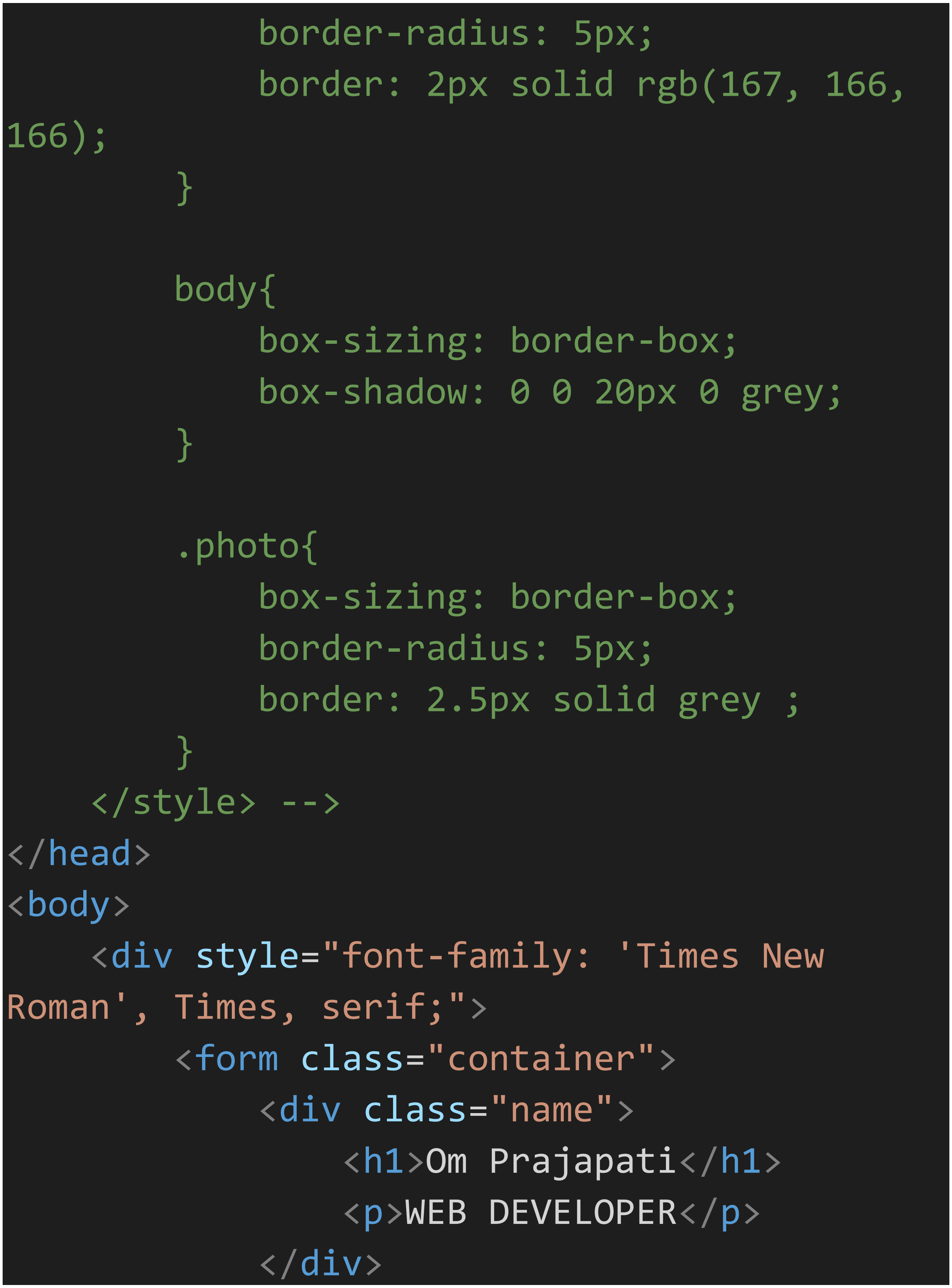
Practical :- 18

AIM :- Make a Resume using the HTML tags without CSS.

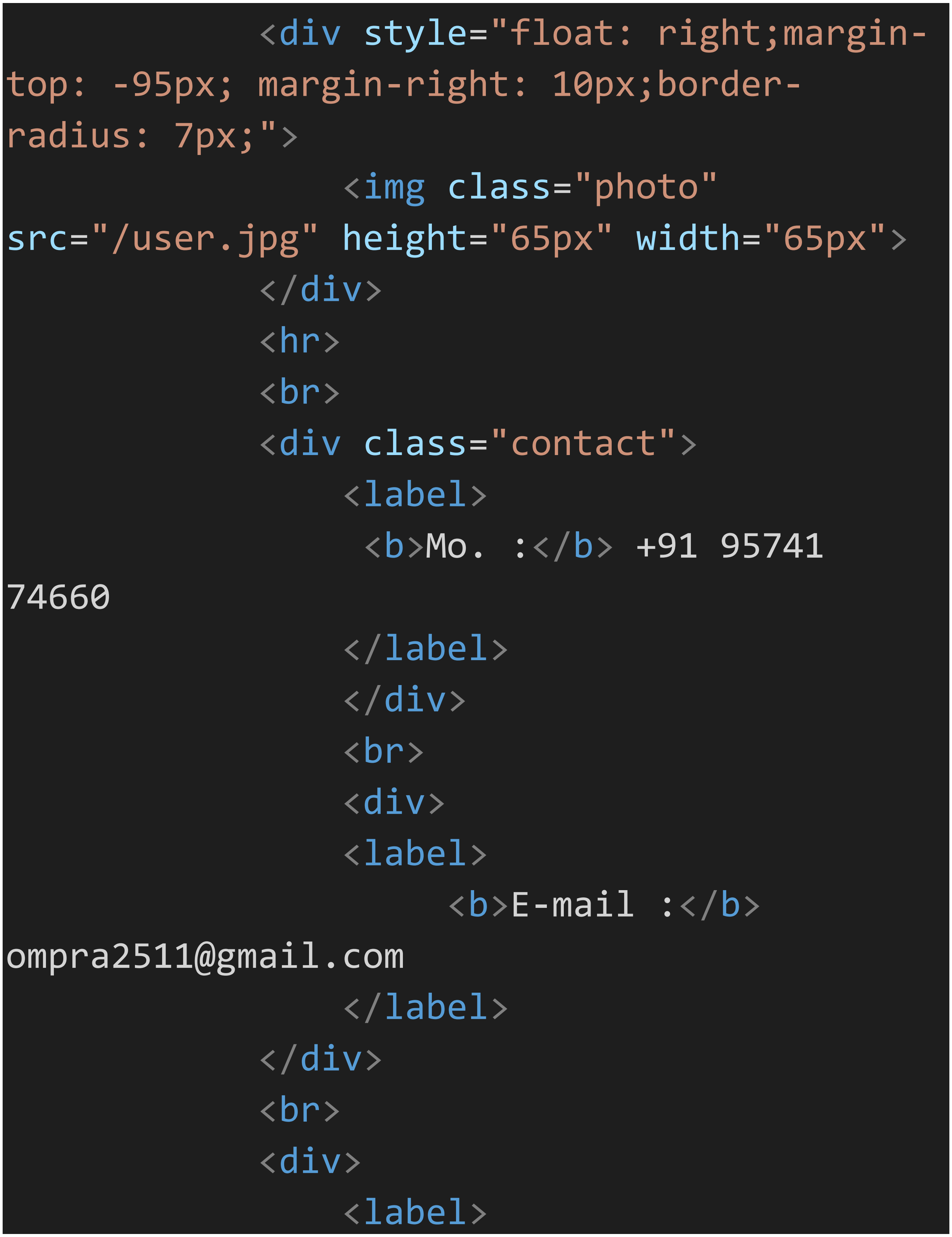
Code :-



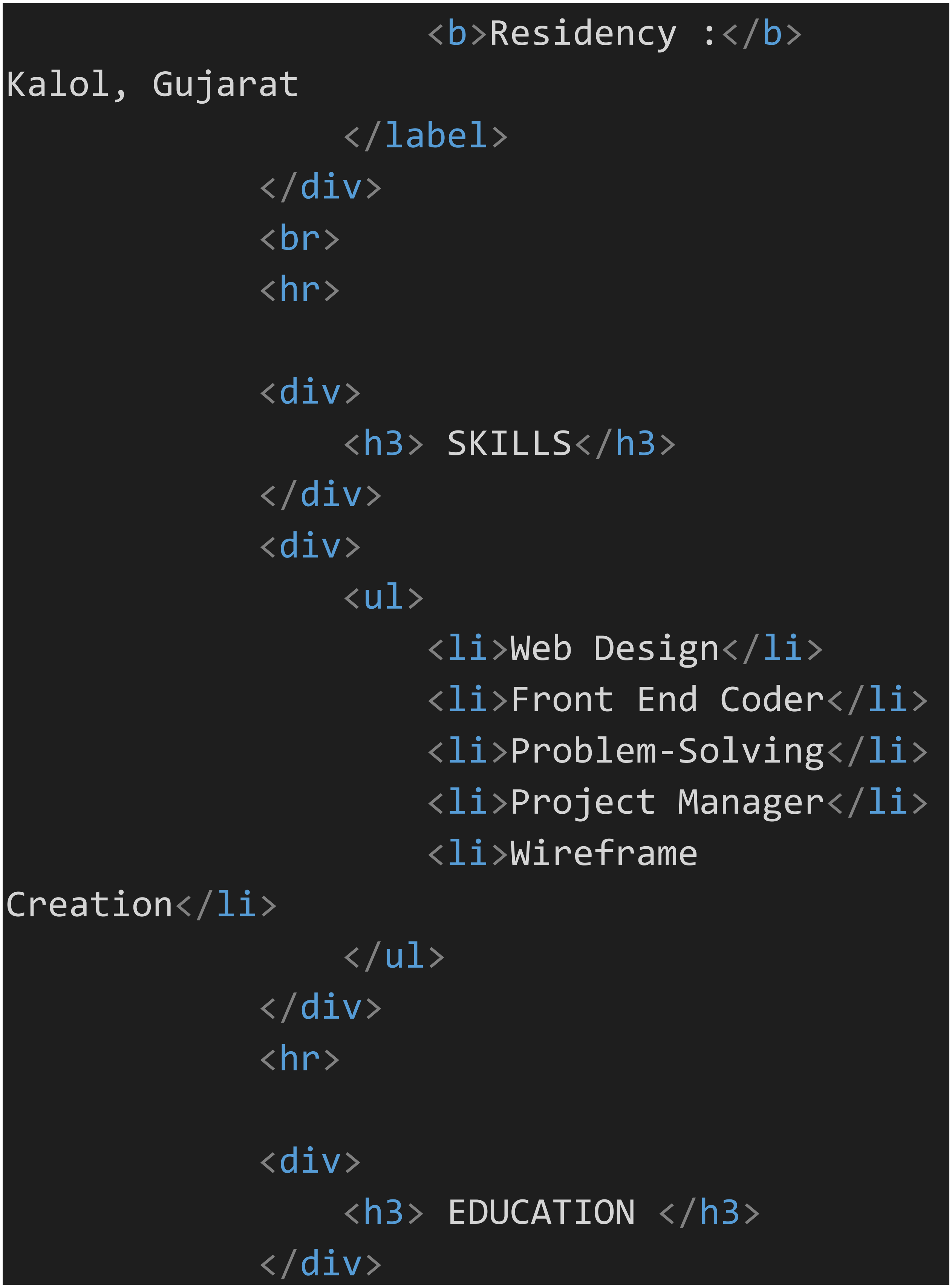
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>My Resume</title>

    <!-- <style>
        body{
            margin: 0 auto;
            max-width: 900px;
            width: 90%;
            padding: 15px;
            margin-top: 15px;
            margin-bottom: 15px;
```



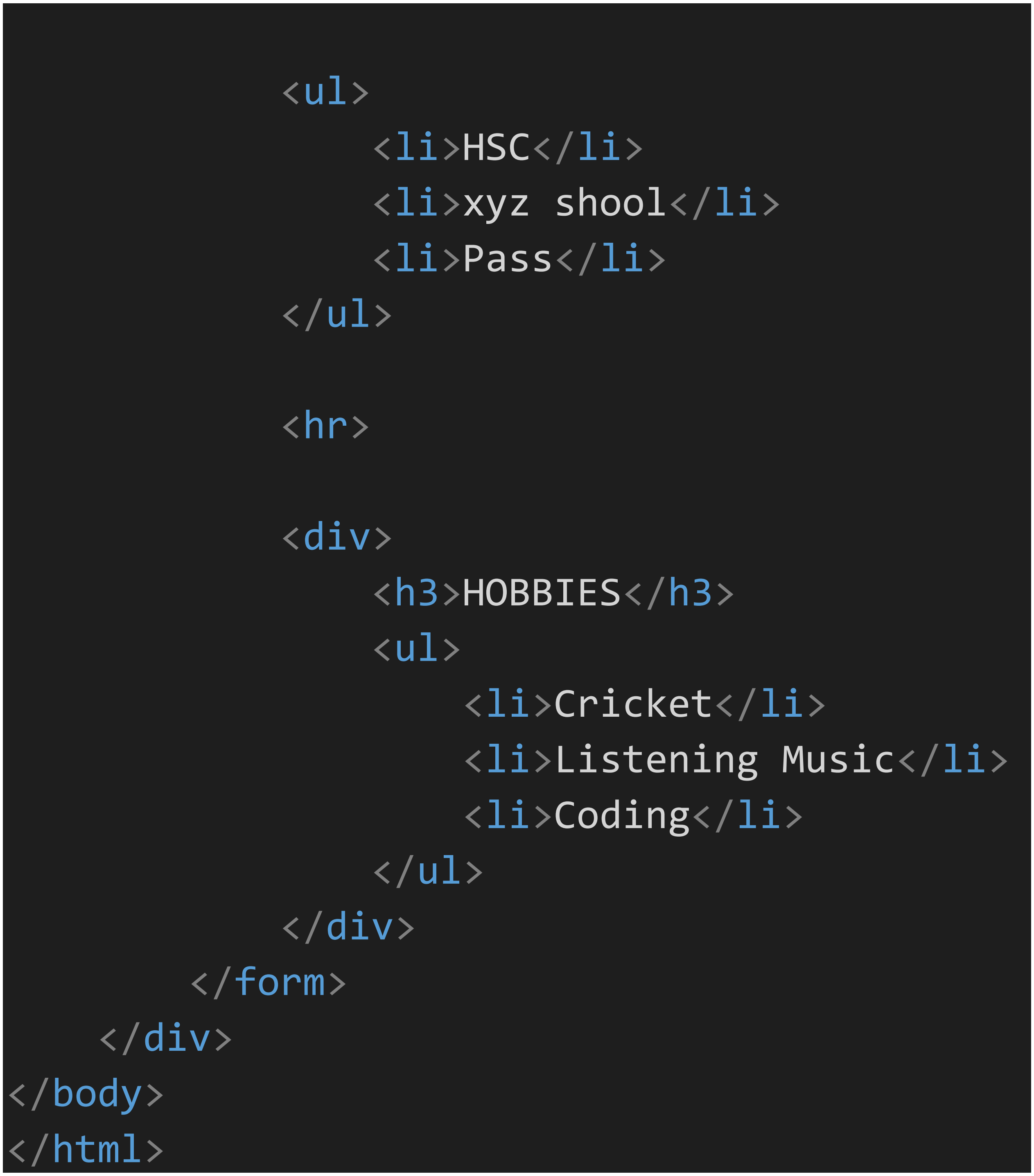
```
            border-radius: 5px;
            border: 2px solid rgb(167, 166,
166);
        }

        body{
            box-sizing: border-box;
            box-shadow: 0 0 20px 0 grey;
        }

        .photo{
            box-sizing: border-box;
            border-radius: 5px;
            border: 2.5px solid grey ;
        }
    </style> -->
</head>
<body>
    <div style="font-family: 'Times New
Roman', Times, serif;">
        <form class="container">
            <div class="name">
                <h1>Om Prajapati</h1>
                <p>WEB DEVELOPER</p>
            </div>
```



```
            <div style="float: right;margin-
top: -95px; margin-right: 10px;border-
radius: 7px;">
                <img class="photo"
src="/user.jpg" height="65px" width="65px">
            </div>
            <hr>
            <br>
            <div class="contact">
                <label>
                 <b>Mo. :</b> +91 95741
74660
                </label>
                </div>
                <br>
                <div>
                <label>
                     <b>E-mail :</b>
ompra2511@gmail.com
                </label>
            </div>
            <br>
            <div>
                <label>
```



```
                    <b>Residency :</b>
Kalol, Gujarat
                </label>
            </div>
            <br>
            <hr>

            <div>
                <h3> SKILLS</h3>
            </div>
            <div>
                <ul>
                    <li>Web Design</li>
                    <li>Front End Coder</li>
                    <li>Problem-Solving</li>
                    <li>Project Manager</li>
                    <li>Wireframe
Creation</li>
                </ul>
            </div>
            <hr>

            <div>
                <h3> EDUCATION </h3>
            </div>
```



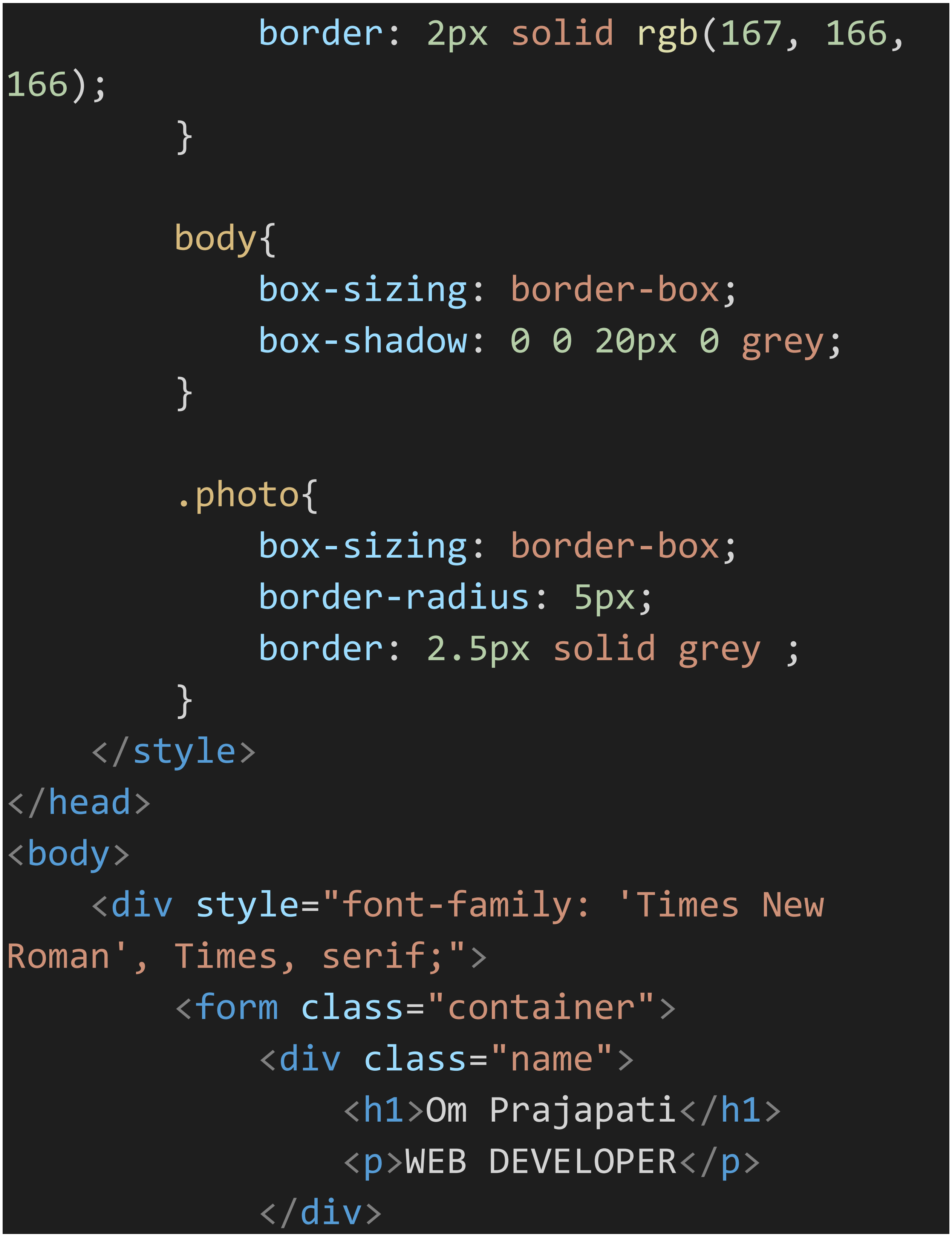
```
            <ul>
                <li>HSC</li>
                <li>xyz shool</li>
                <li>Pass</li>
            </ul>

            <hr>

            <div>
                <h3>HOBBIES</h3>
                <ul>
                    <li>Cricket</li>
                    <li>Listening Music</li>
                    <li>Coding</li>
                </ul>
            </div>
        </form>
    </div>
</body>
</html>
```

Practical :- 22

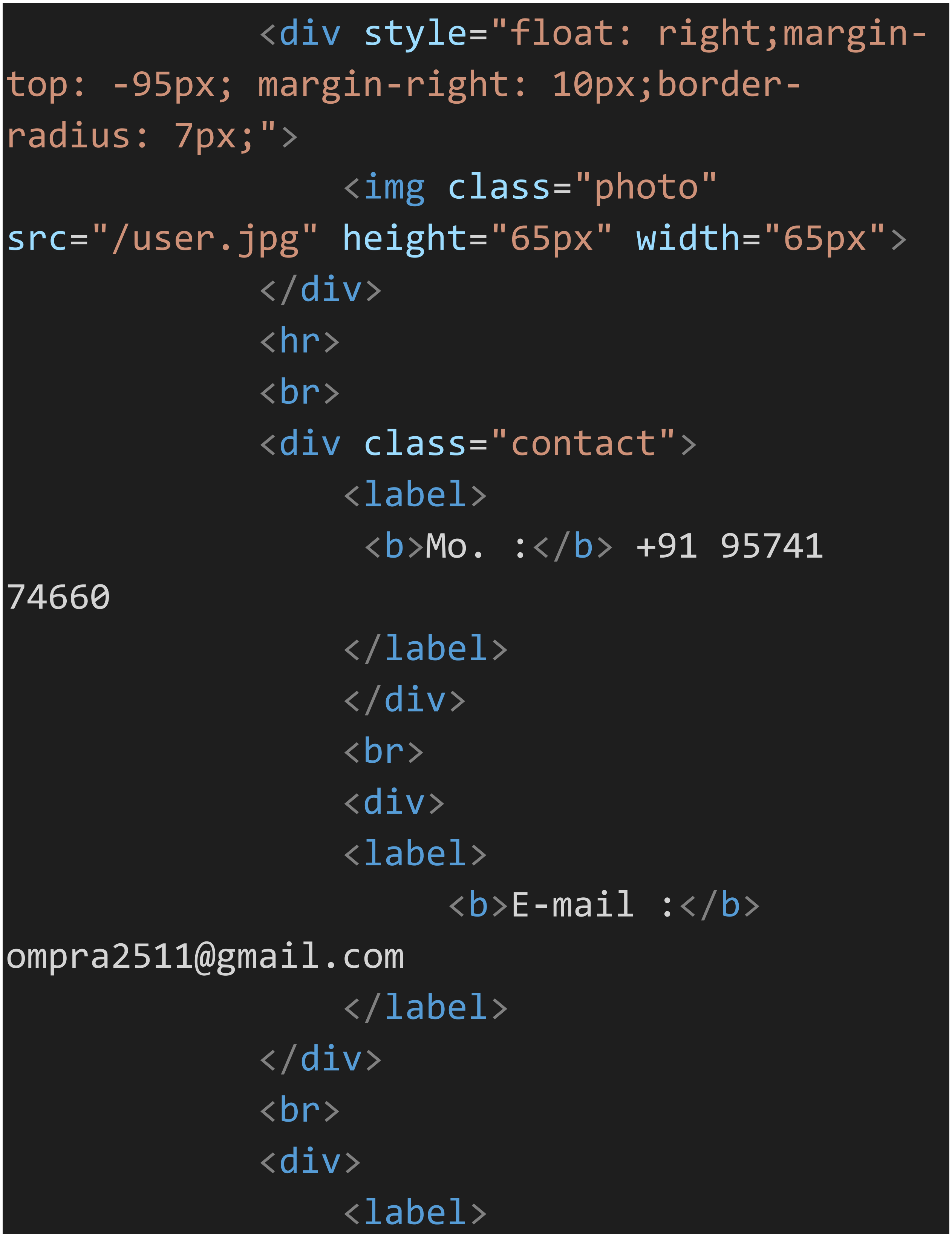
AIM :- Make a Resume using the HTML tags with CSS.

Code :-

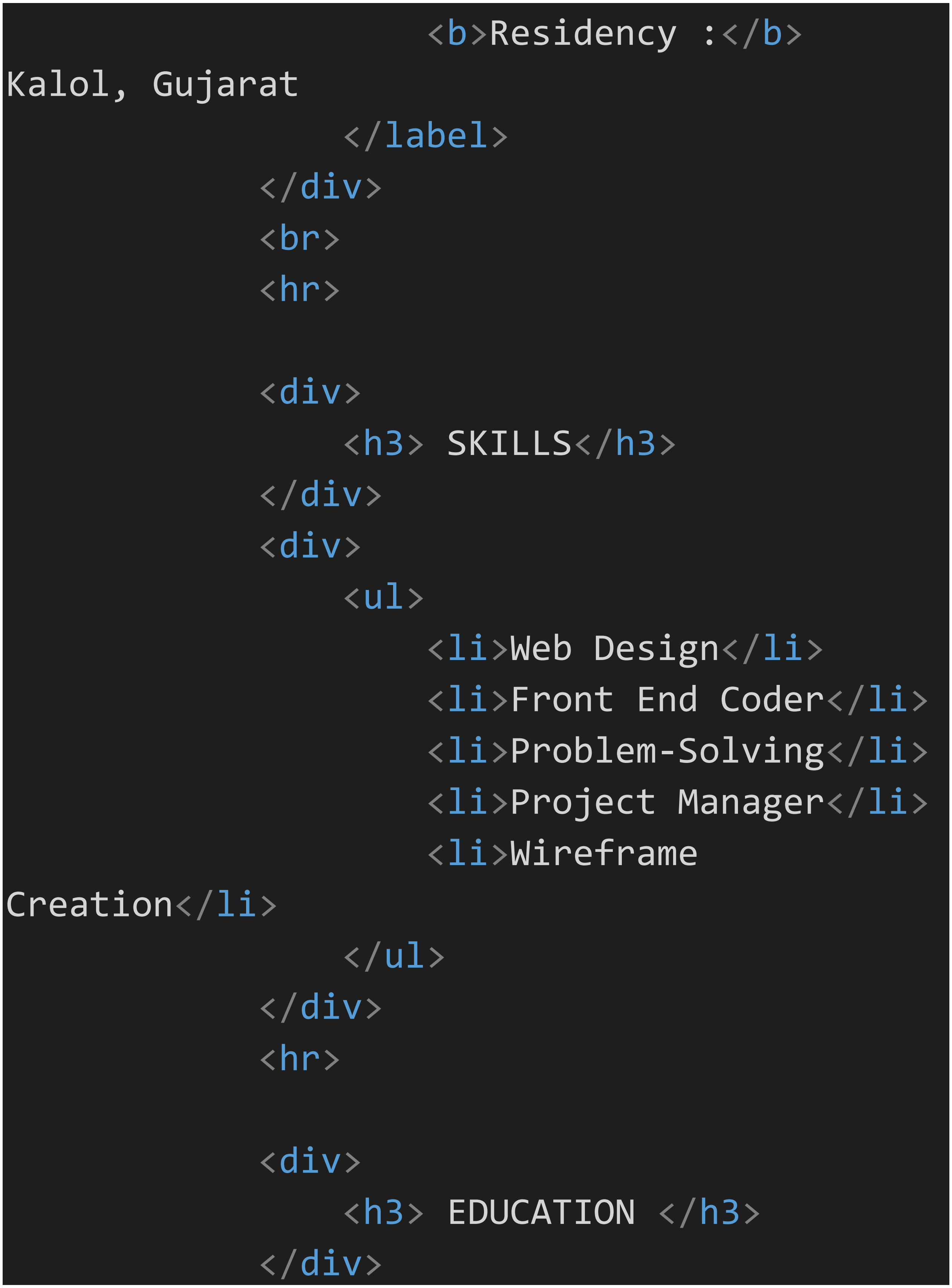
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Resume</title>

    <style>
        body{
            margin: 0 auto;
            max-width: 900px;
            width: 90%;
            padding: 15px;
            margin-top: 15px;
            margin-bottom: 15px;
            border-radius: 5px;
```



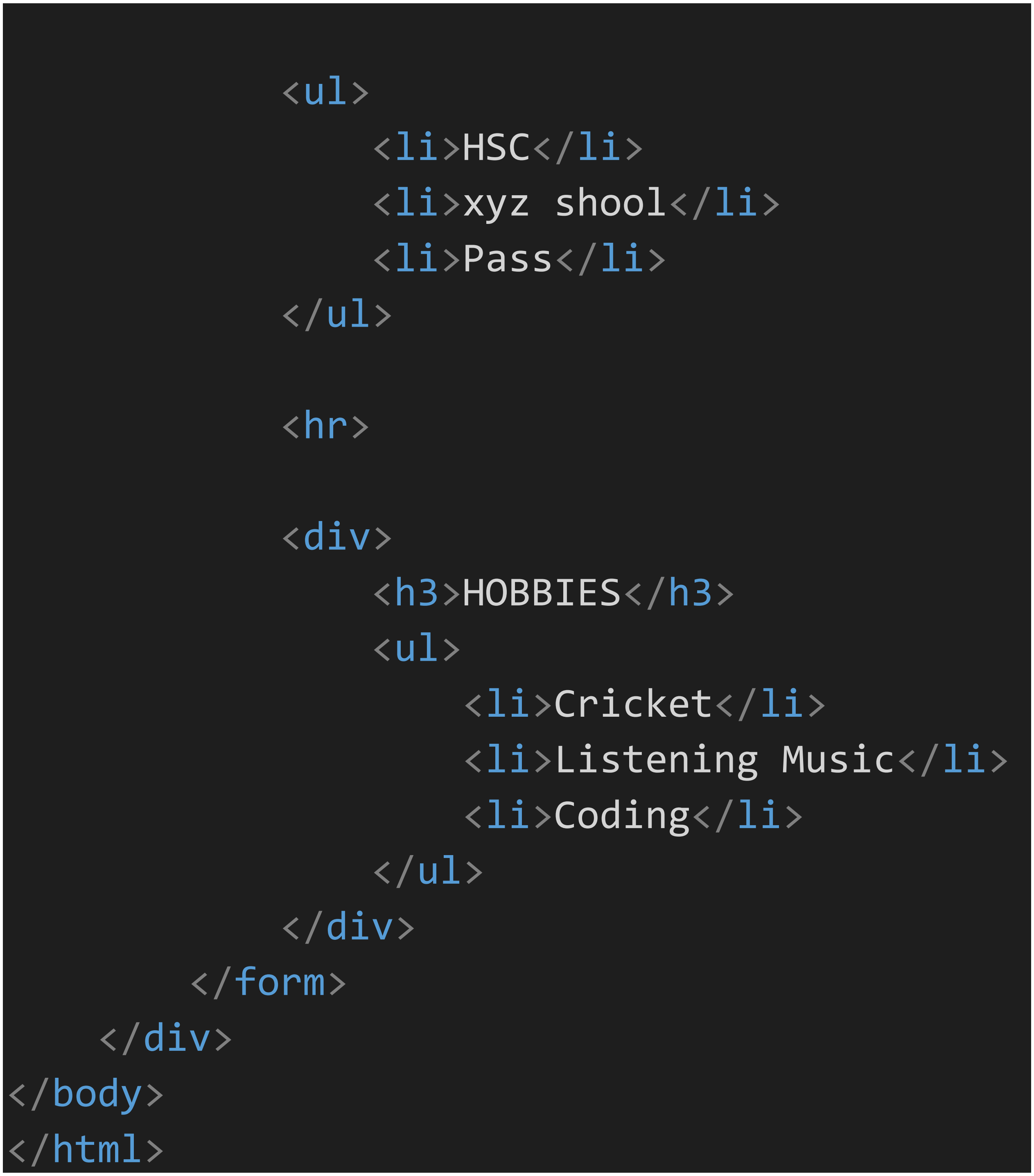
```
            border: 2px solid rgb(167, 166,
166);
        }

        body{
            box-sizing: border-box;
            box-shadow: 0 0 20px 0 grey;
        }

        .photo{
            box-sizing: border-box;
            border-radius: 5px;
            border: 2.5px solid grey ;
        }
    </style>
</head>
<body>
    <div style="font-family: 'Times New
Roman', Times, serif;">
        <form class="container">
            <div class="name">
                <h1>Om Prajapati</h1>
                <p>WEB DEVELOPER</p>
            </div>
```



```
            <div style="float: right;margin-
top: -95px; margin-right: 10px;border-
radius: 7px;">
                <img class="photo"
src="/user.jpg" height="65px" width="65px">
            </div>
            <hr>
            <br>
            <div class="contact">
                <label>
                 <b>Mo. :</b> +91 95741
74660
                </label>
                </div>
                <br>
                <div>
                <label>
                     <b>E-mail :</b>
ompra2511@gmail.com
                </label>
            </div>
            <br>
            <div>
                <label>
```



```
                    <b>Residency :</b>
Kalol, Gujarat
                </label>
            </div>
            <br>
            <hr>

            <div>
                <h3> SKILLS</h3>
            </div>
            <div>
                <ul>
                    <li>Web Design</li>
                    <li>Front End Coder</li>
                    <li>Problem-Solving</li>
                    <li>Project Manager</li>
                    <li>Wireframe
Creation</li>
                </ul>
            </div>
            <hr>

            <div>
                <h3> EDUCATION </h3>
            </div>
```



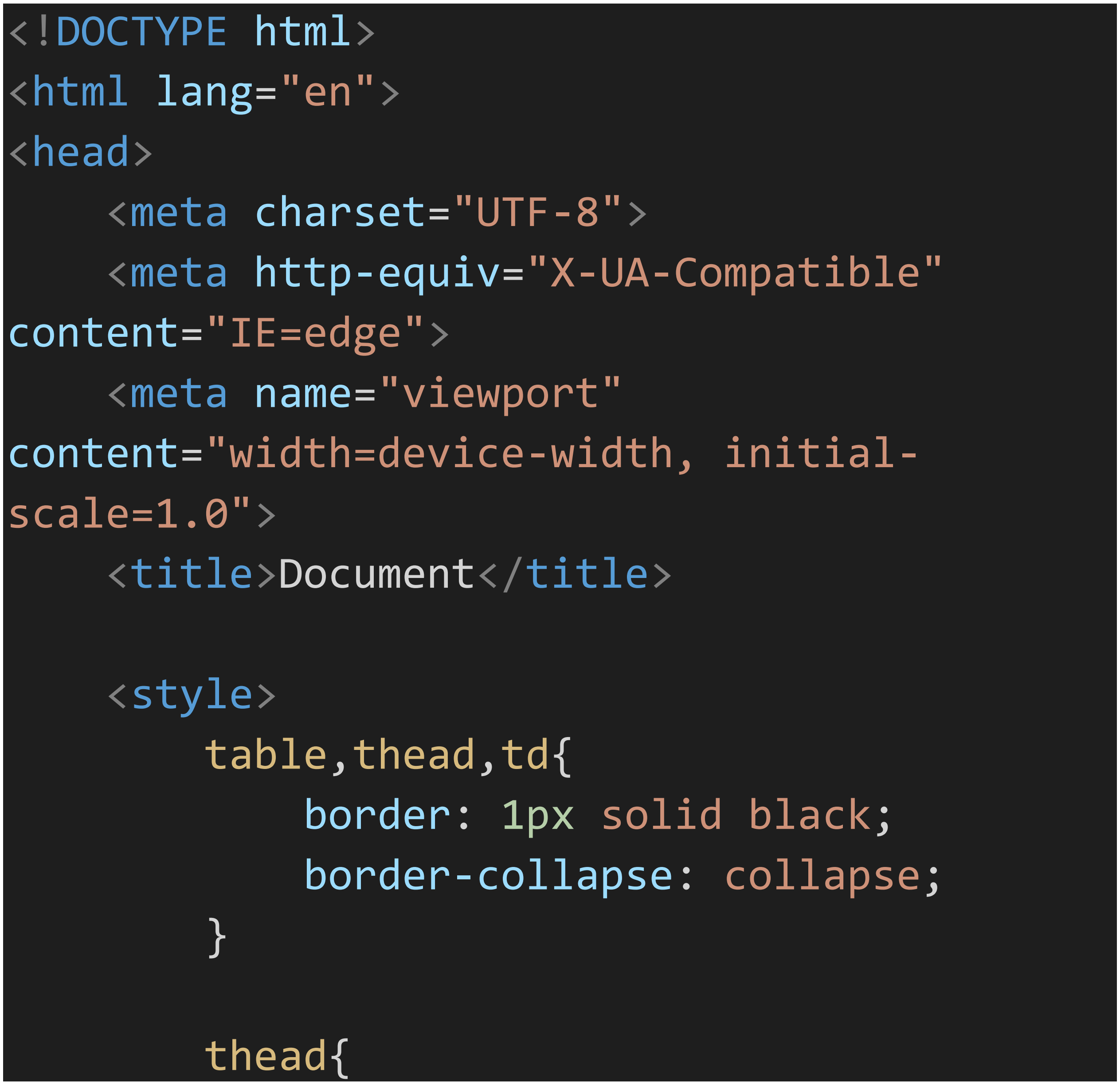
```html
            <ul>
                <li>HSC</li>
                <li>xyz shool</li>
                <li>Pass</li>
            </ul>

            <hr>

            <div>
                <h3>HOBBIES</h3>
                <ul>
                    <li>Cricket</li>
                    <li>Listening Music</li>
                    <li>Coding</li>
                </ul>
            </div>
        </form>
    </div>
</body>
</html>
```

Practical :- 20

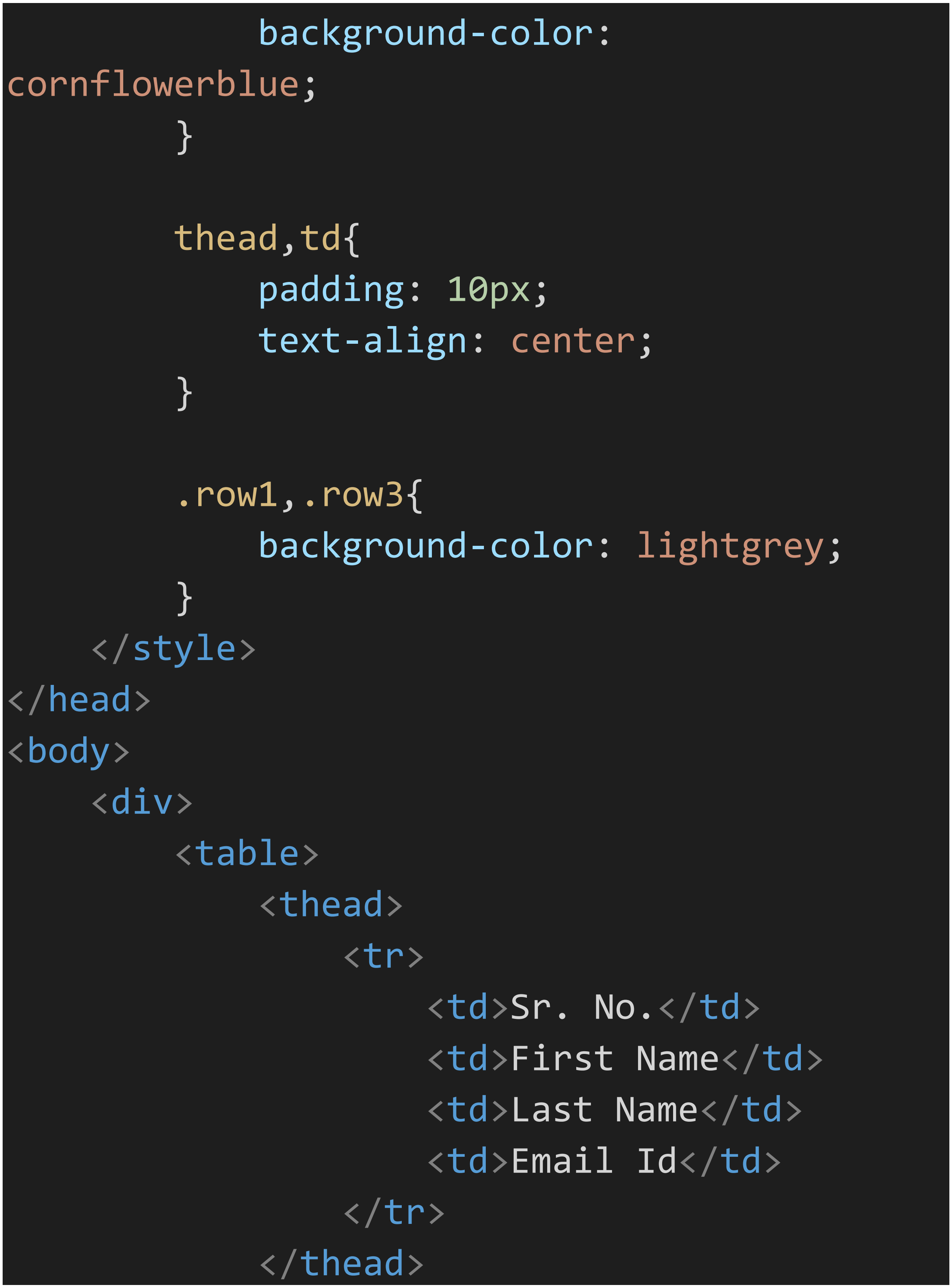
AIM :- Create an HTML page table and form.

Code :-

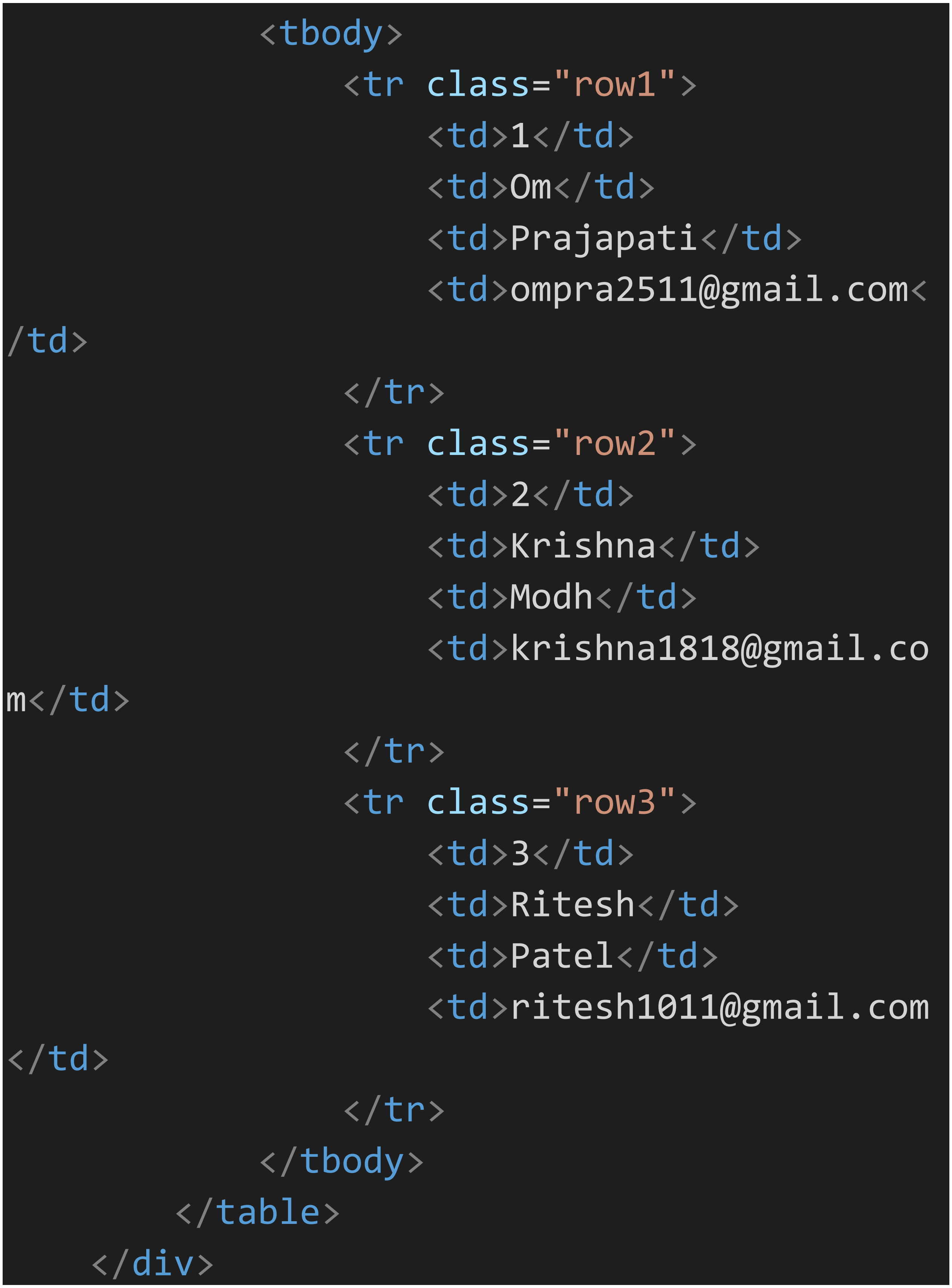
Table



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <style>
        table,thead,td{
            border: 1px solid black;
            border-collapse: collapse;
        }

        thead{
```



```
            background-color:
cornflowerblue;
        }

        thead,td{
            padding: 10px;
            text-align: center;
        }

        .row1,.row3{
            background-color: lightgrey;
        }
    </style>
</head>
<body>
    <div>
        <table>
            <thead>
                <tr>
                    <td>Sr. No.</td>
                    <td>First Name</td>
                    <td>Last Name</td>
                    <td>Email Id</td>
                </tr>
            </thead>
```

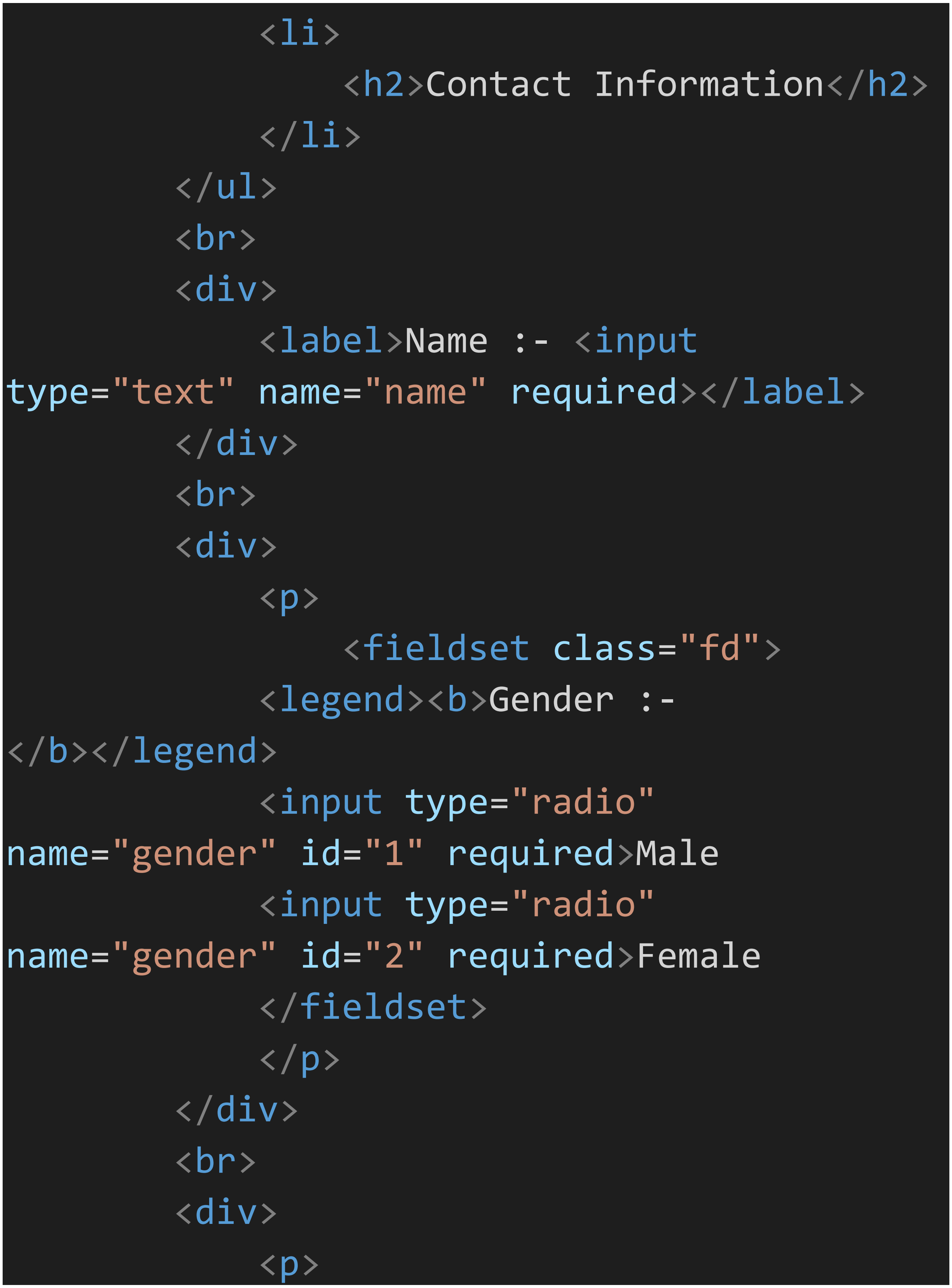


```
            <tbody>
                <tr class="row1">
                    <td>1</td>
                    <td>Om</td>
                    <td>Prajapati</td>
                    <td>ompra2511@gmail.com<
/td>
                </tr>
                <tr class="row2">
                    <td>2</td>
                    <td>Krishna</td>
                    <td>Modh</td>
                    <td>krishna1818@gmail.co
m</td>
                </tr>
                <tr class="row3">
                    <td>3</td>
                    <td>Ritesh</td>
                    <td>Patel</td>
                    <td>ritesh1011@gmail.com
</td>
                </tr>
            </tbody>
        </table>
    </div>
```

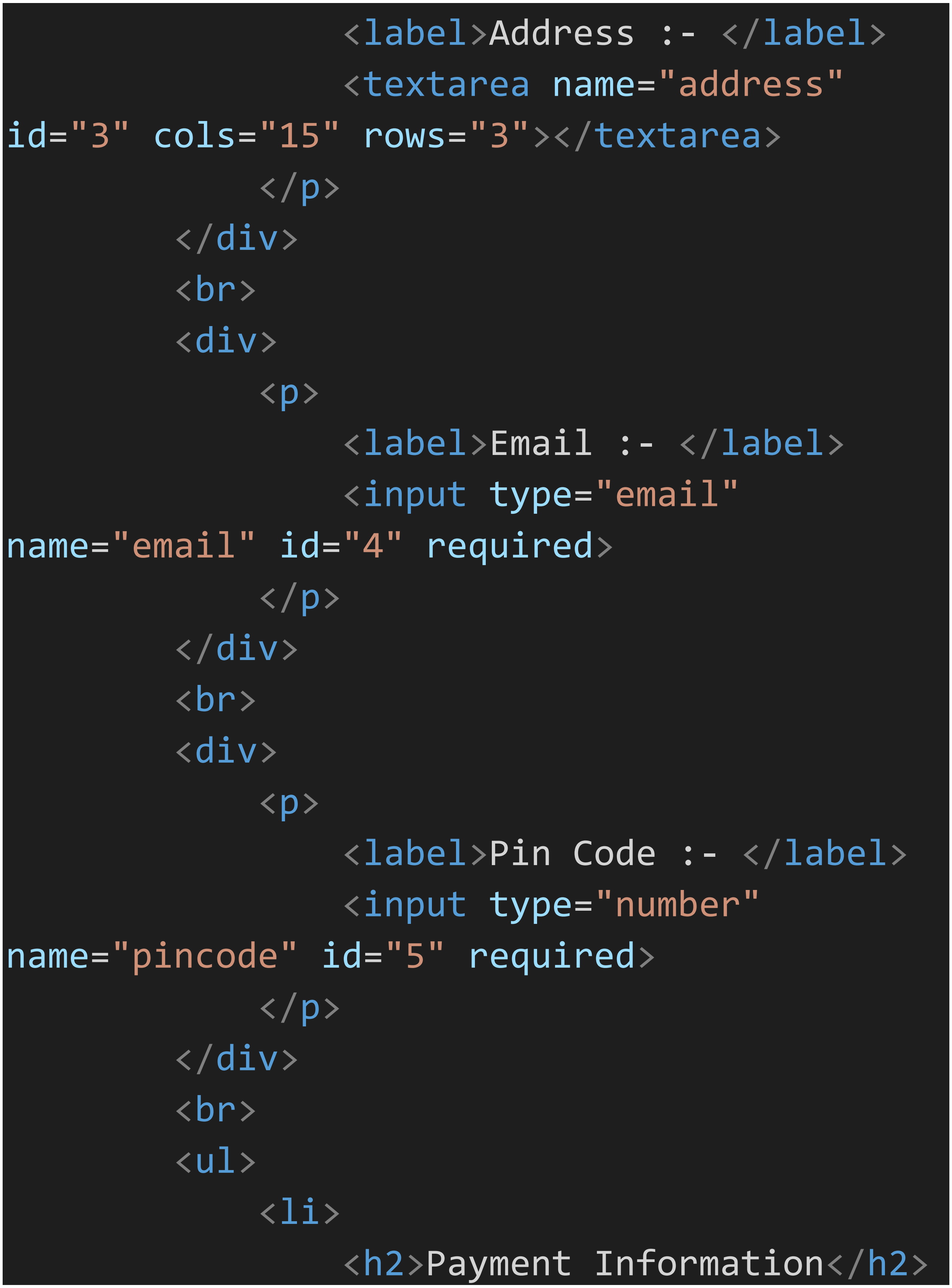
```
</body>
</html>
```

Form

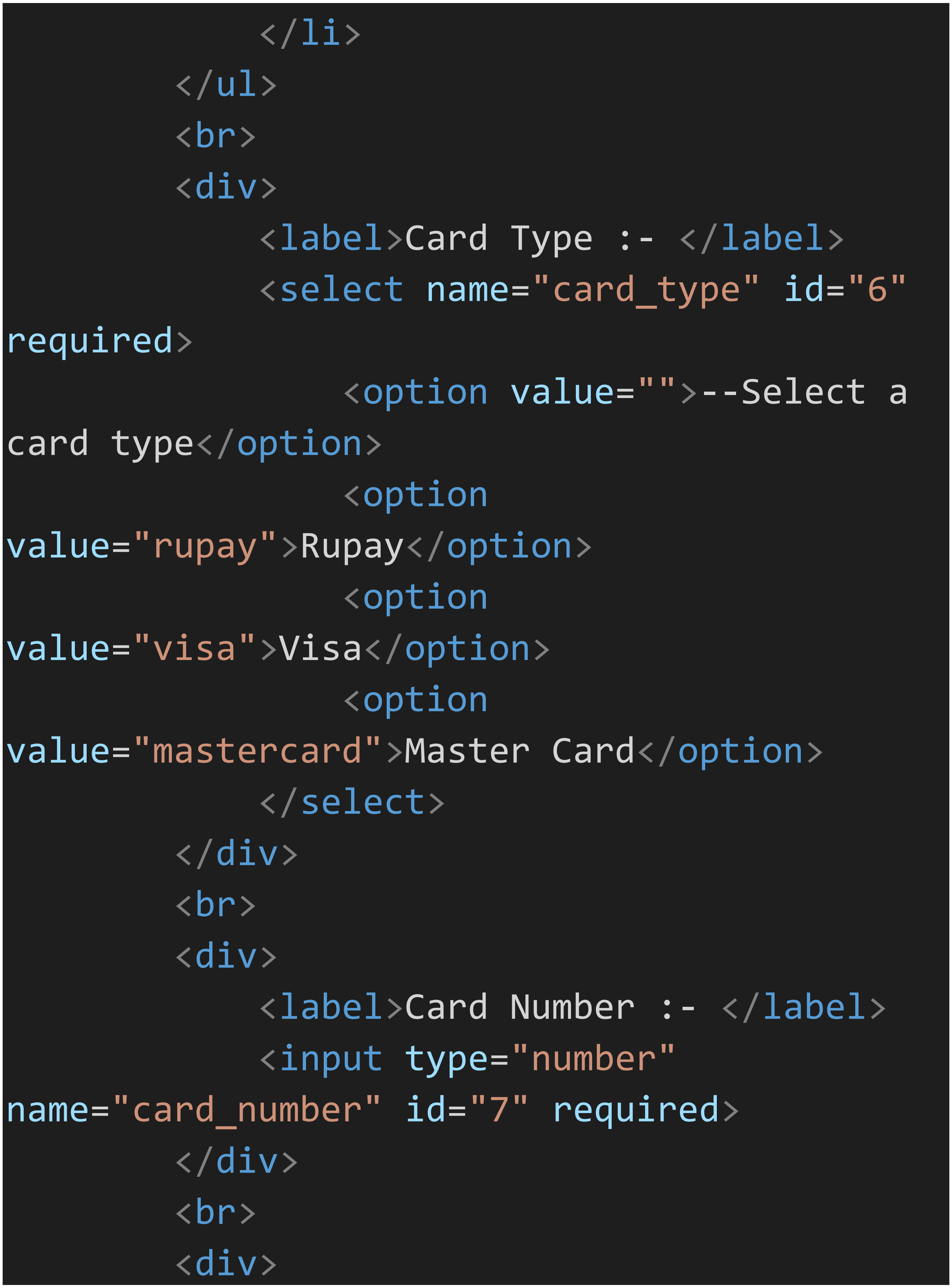
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Payment Form</title>
    <link rel="stylesheet" type="text/css"
href="/style.css">
</head>
<body>
    <div class="container">
    <form>
        <h1
class="main_heading"><b><center>Payment
Form</center></b></h1>
        <hr>
        <ul>
```



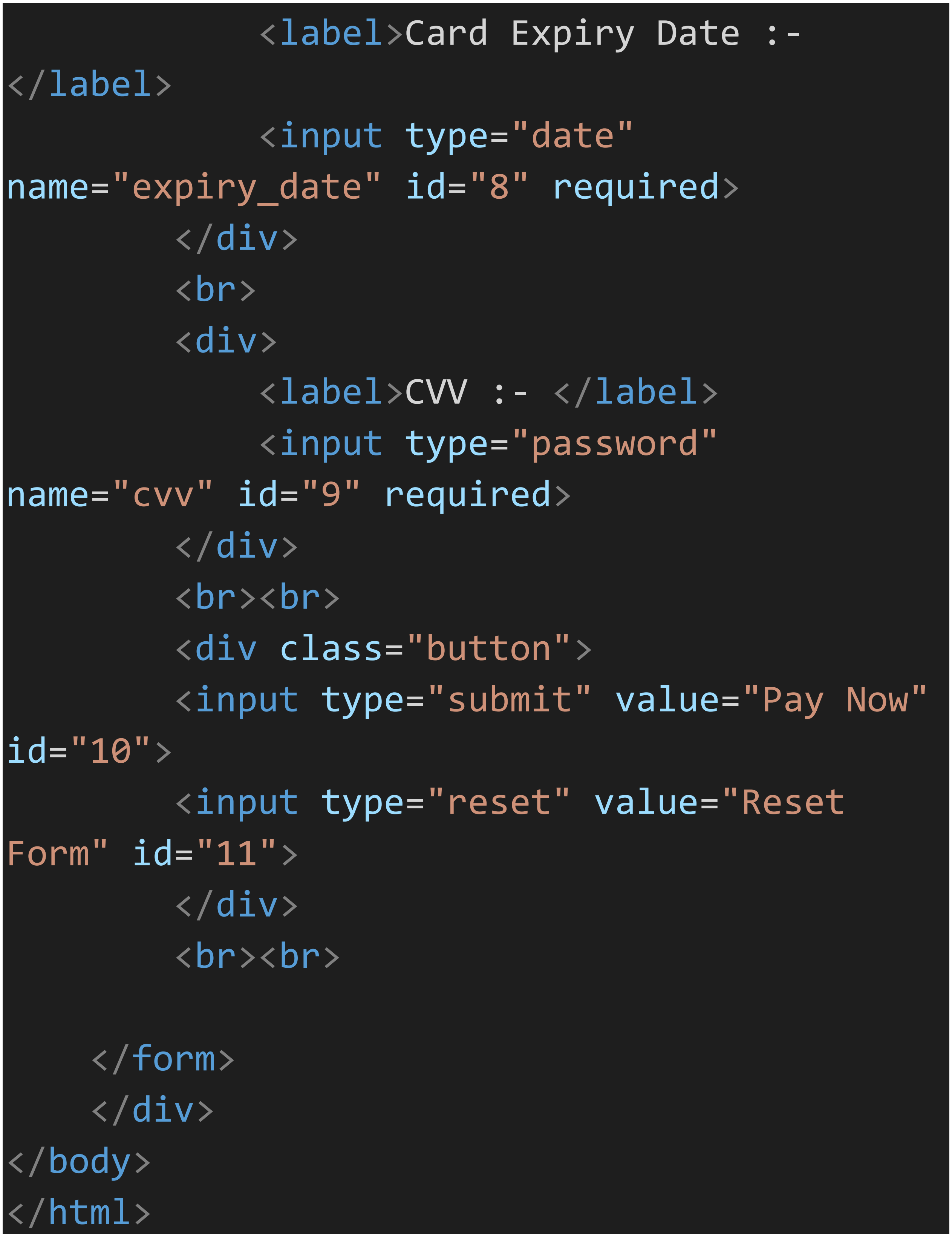
```
            <li>
                <h2>Contact Information</h2>
            </li>
        </ul>
        <br>
        <div>
            <label>Name :- <input 
type="text" name="name" required></label>
        </div>
        <br>
        <div>
            <p>
                <fieldset class="fd">
            <legend><b>Gender :- 
</b></legend>
            <input type="radio" 
name="gender" id="1" required>Male
            <input type="radio" 
name="gender" id="2" required>Female
            </fieldset>
            </p>
        </div>
        <br>
        <div>
            <p>
```



```
                <label>Address :- </label>
                <textarea name="address"
id="3" cols="15" rows="3"></textarea>
            </p>
        </div>
        <br>
        <div>
            <p>
                <label>Email :- </label>
                <input type="email"
name="email" id="4" required>
            </p>
        </div>
        <br>
        <div>
            <p>
                <label>Pin Code :- </label>
                <input type="number"
name="pincode" id="5" required>
            </p>
        </div>
        <br>
        <ul>
            <li>
                <h2>Payment Information</h2>
```



```
            </li>
        </ul>
        <br>
        <div>
            <label>Card Type :- </label>
            <select name="card_type" id="6"
required>
                <option value="">--Select a
card type</option>
                <option
value="rupay">Rupay</option>
                <option
value="visa">Visa</option>
                <option
value="mastercard">Master Card</option>
            </select>
        </div>
        <br>
        <div>
            <label>Card Number :- </label>
            <input type="number"
name="card_number" id="7" required>
        </div>
        <br>
        <div>
```



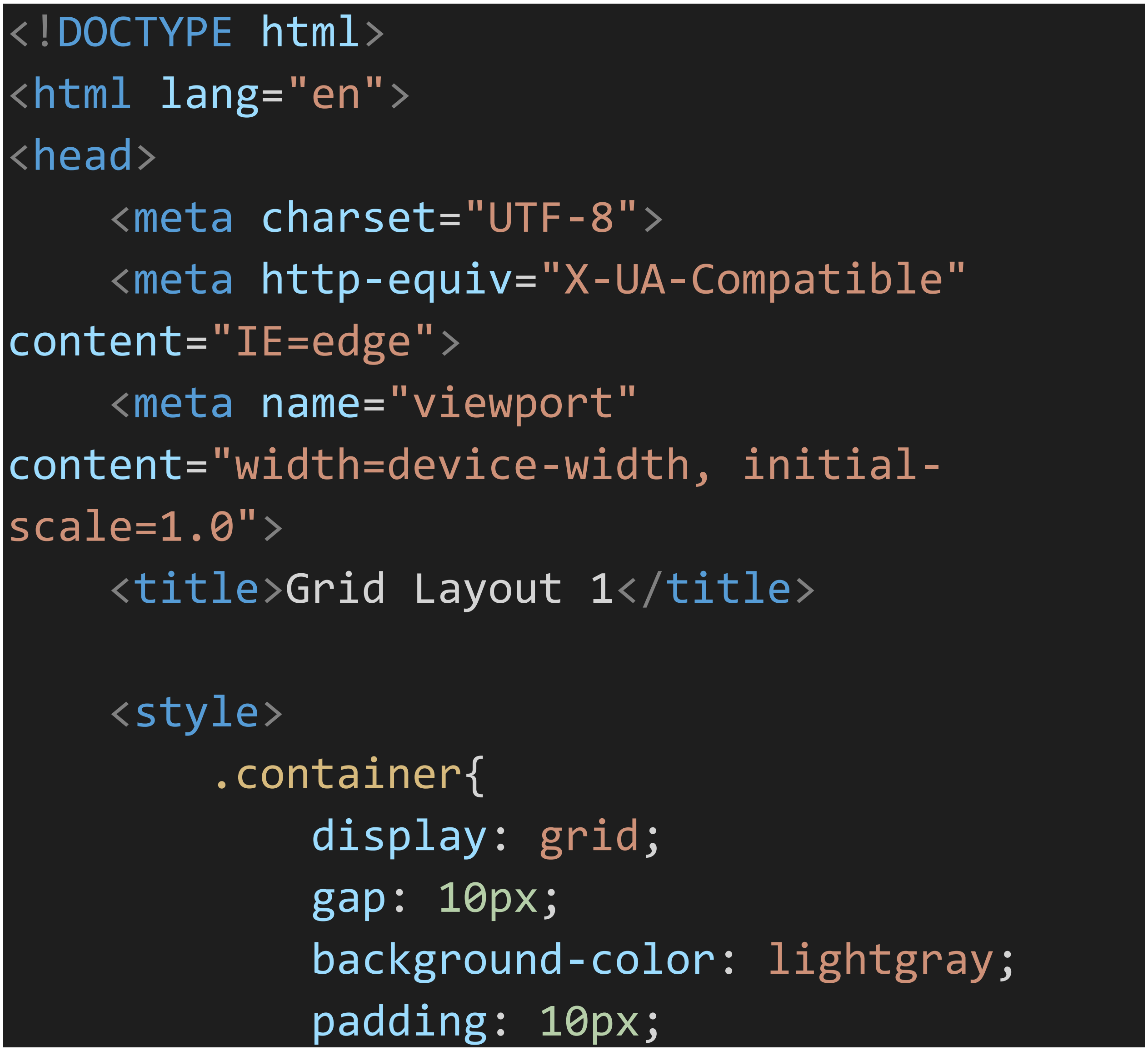
```
            <label>Card Expiry Date :-
</label>
            <input type="date"
name="expiry_date" id="8" required>
        </div>
        <br>
        <div>
            <label>CVV :- </label>
            <input type="password"
name="cvv" id="9" required>
        </div>
        <br><br>
        <div class="button">
        <input type="submit" value="Pay Now"
id="10">
        <input type="reset" value="Reset
Form" id="11">
        </div>
        <br><br>

    </form>
    </div>
</body>
</html>
```

Practical :- 23

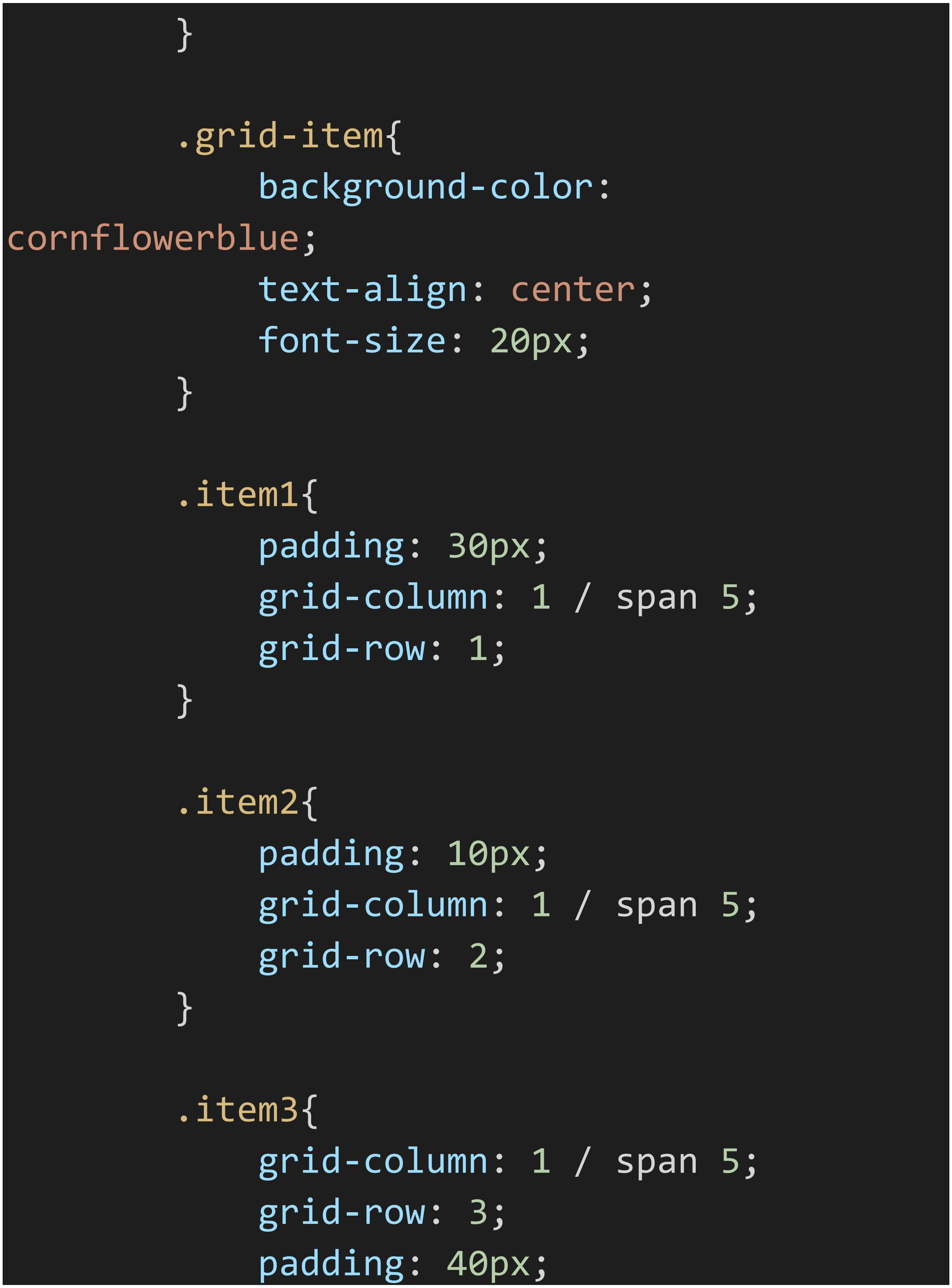
AIM :- Create an HTML Page containing the following Gray Layout using CSS.

Code :-

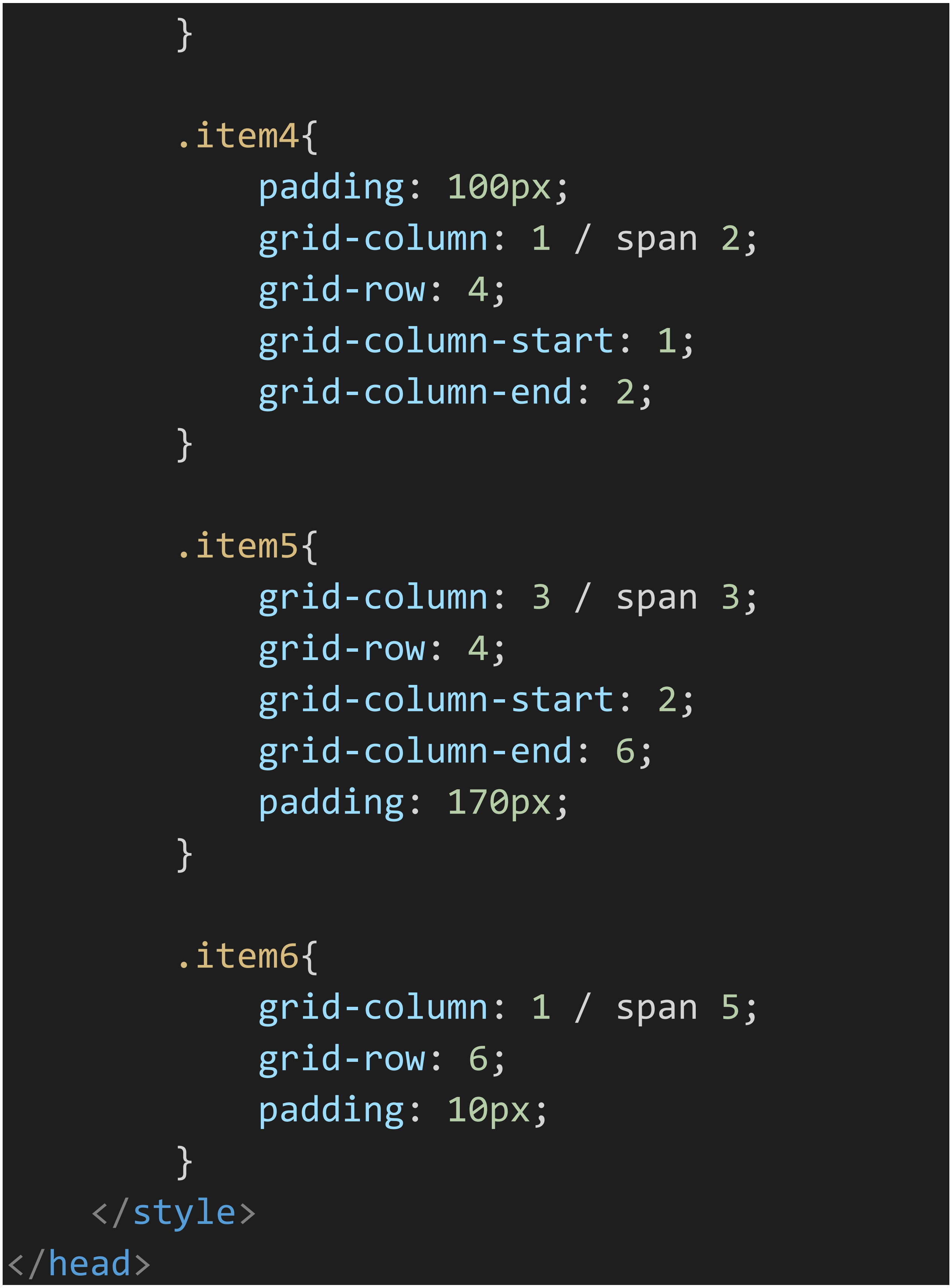
Layout 1



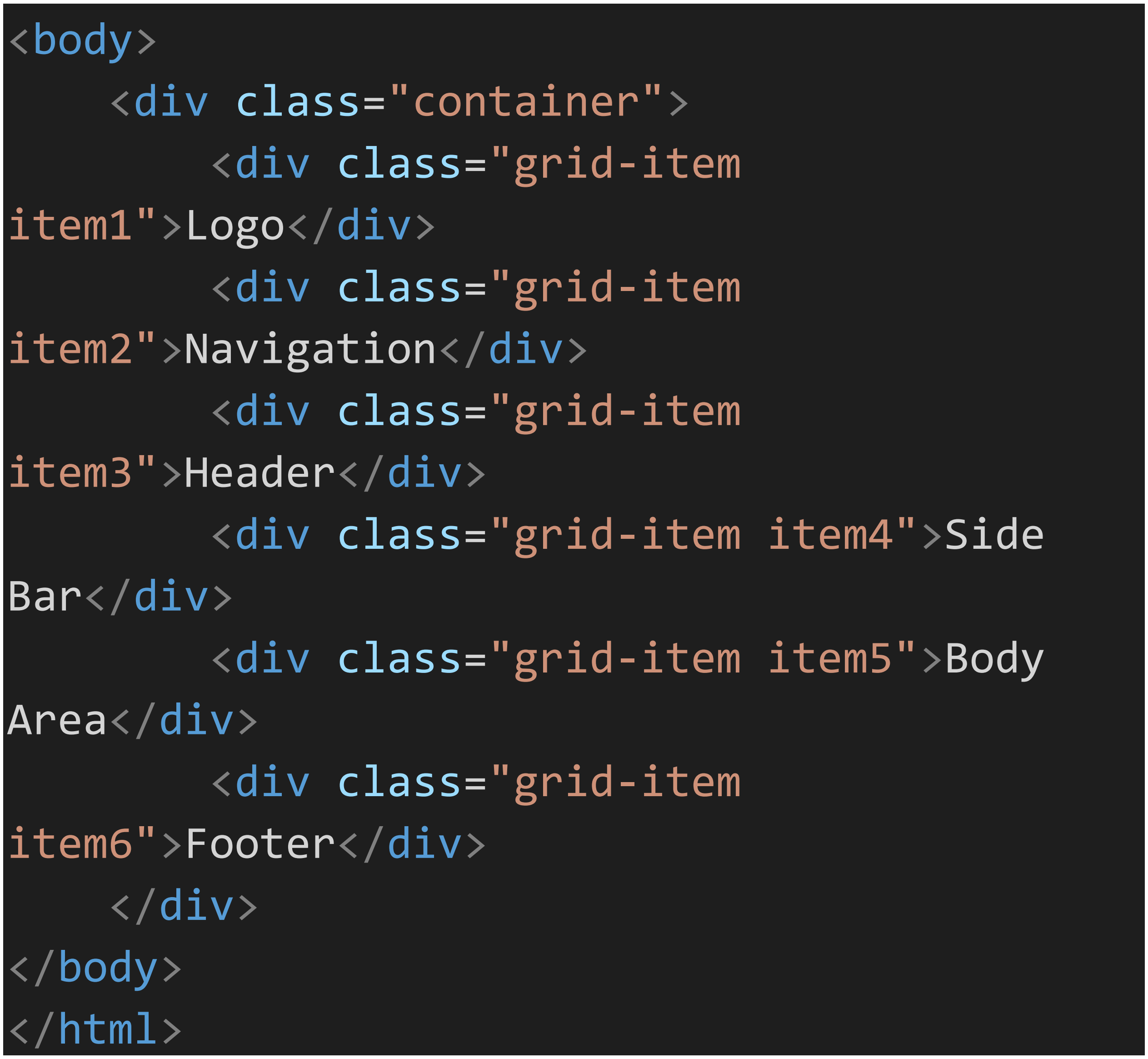
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Grid Layout 1</title>

    <style>
        .container{
            display: grid;
            gap: 10px;
            background-color: lightgray;
            padding: 10px;
```



```
        }

        .grid-item{
            background-color:
cornflowerblue;
            text-align: center;
            font-size: 20px;
        }

        .item1{
            padding: 30px;
            grid-column: 1 / span 5;
            grid-row: 1;
        }

        .item2{
            padding: 10px;
            grid-column: 1 / span 5;
            grid-row: 2;
        }

        .item3{
            grid-column: 1 / span 5;
            grid-row: 3;
            padding: 40px;
```

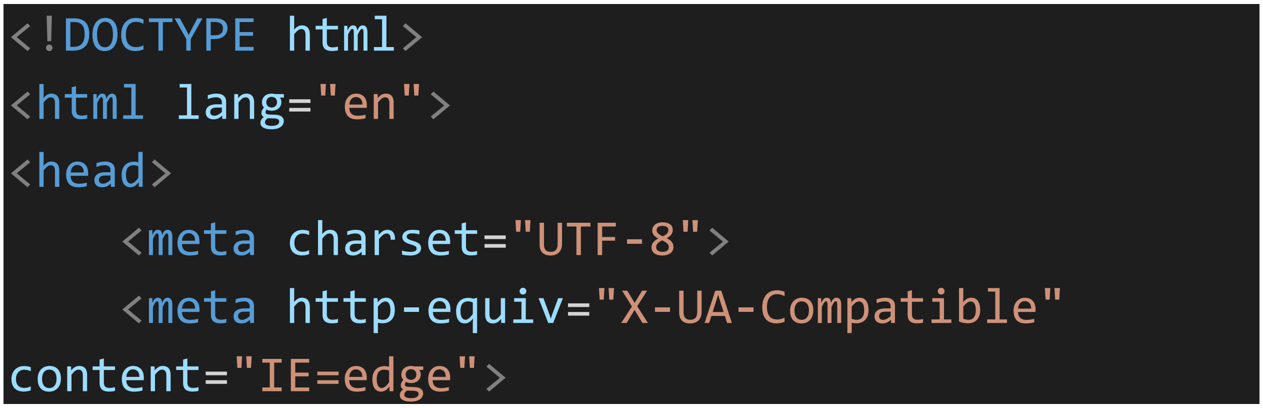


```
        }

        .item4{
            padding: 100px;
            grid-column: 1 / span 2;
            grid-row: 4;
            grid-column-start: 1;
            grid-column-end: 2;
        }

        .item5{
            grid-column: 3 / span 3;
            grid-row: 4;
            grid-column-start: 2;
            grid-column-end: 6;
            padding: 170px;
        }

        .item6{
            grid-column: 1 / span 5;
            grid-row: 6;
            padding: 10px;
        }
    </style>
</head>
```

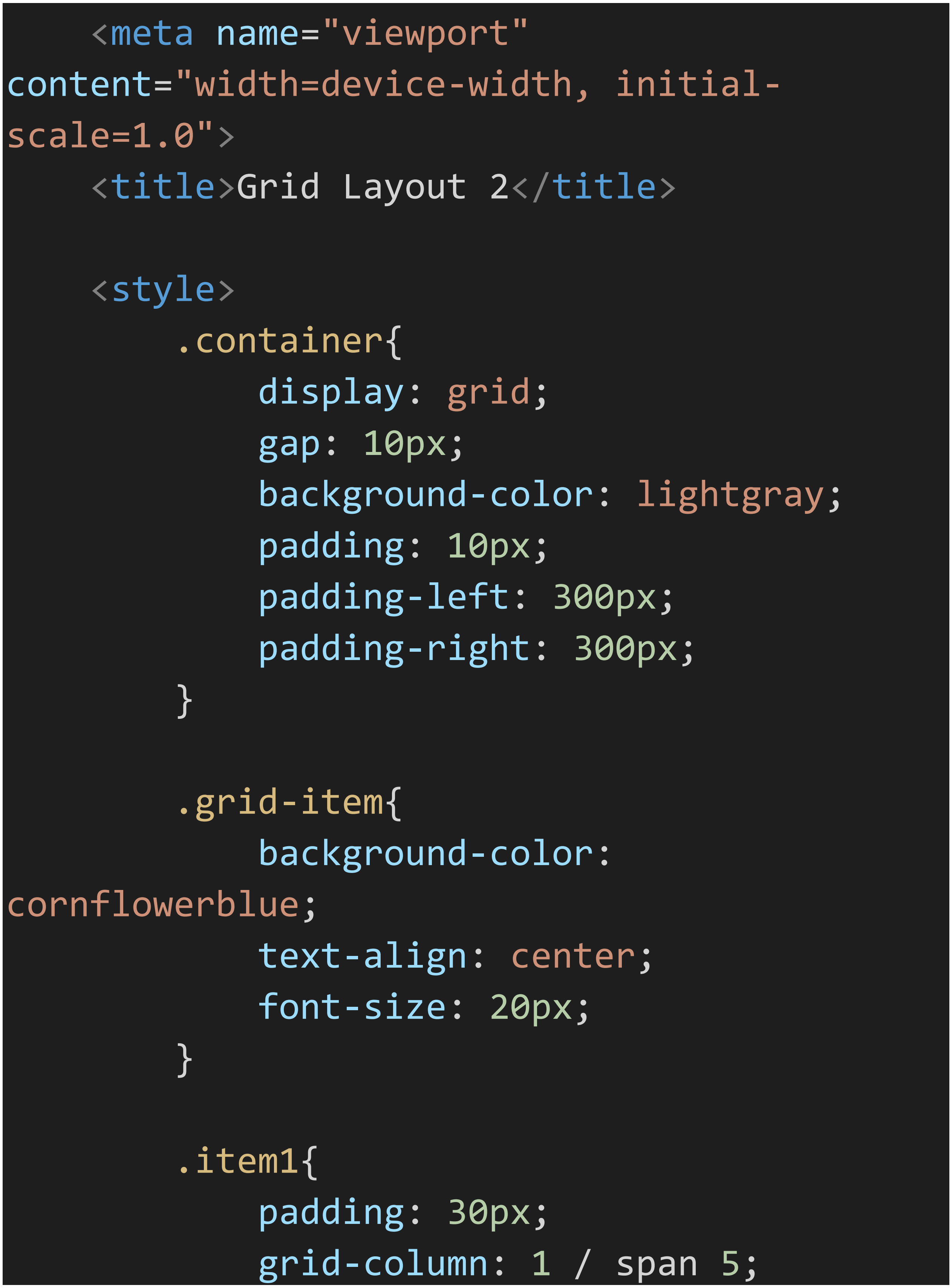


```
<body>
    <div class="container">
        <div class="grid-item
item1">Logo</div>
        <div class="grid-item
item2">Navigation</div>
        <div class="grid-item
item3">Header</div>
        <div class="grid-item item4">Side
Bar</div>
        <div class="grid-item item5">Body
Area</div>
        <div class="grid-item
item6">Footer</div>
    </div>
</body>
</html>
```

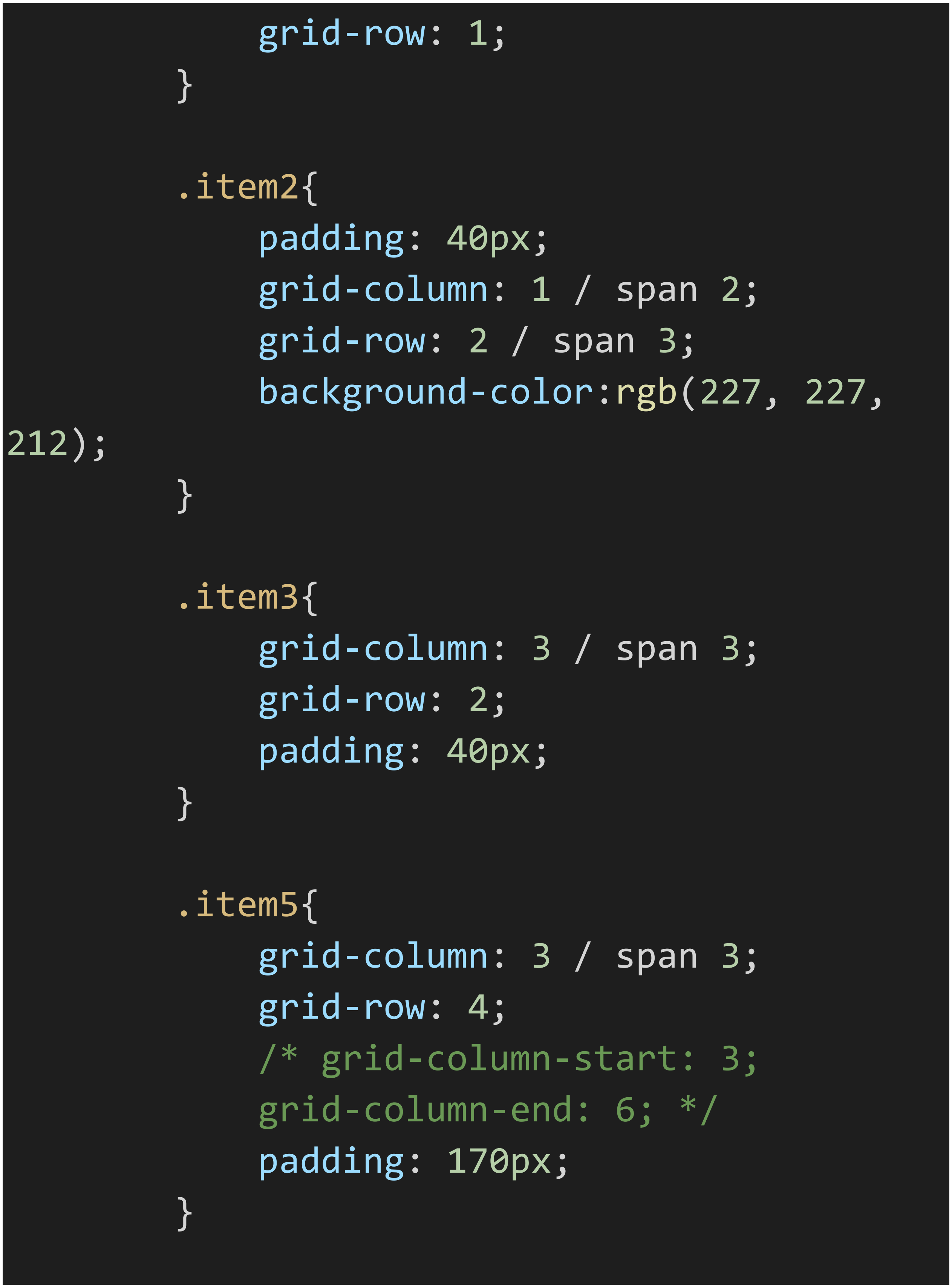
Layout 2



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
```



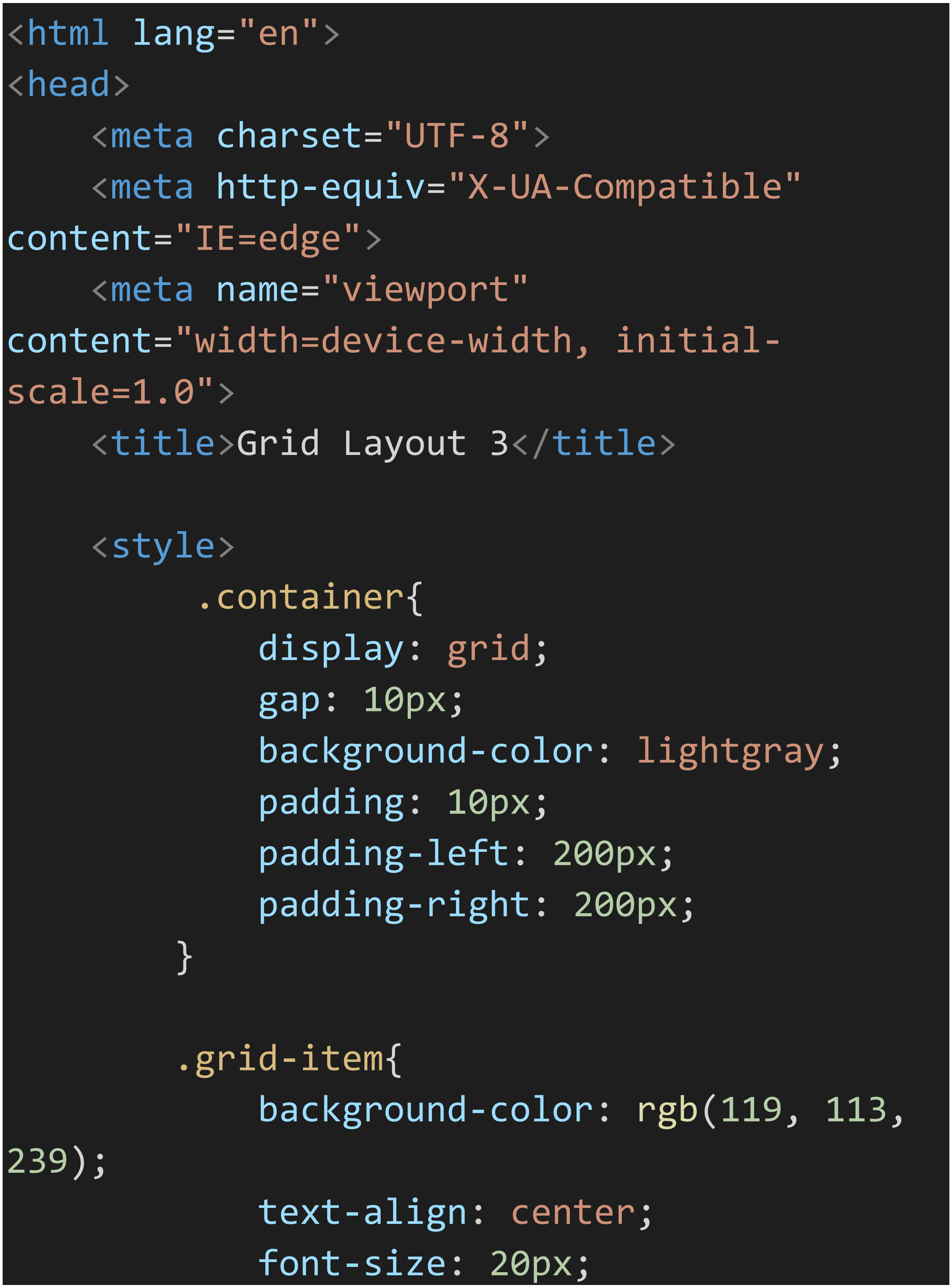
```
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Grid Layout 2</title>

    <style>
        .container{
            display: grid;
            gap: 10px;
            background-color: lightgray;
            padding: 10px;
            padding-left: 300px;
            padding-right: 300px;
        }

        .grid-item{
            background-color:
cornflowerblue;
            text-align: center;
            font-size: 20px;
        }

        .item1{
            padding: 30px;
            grid-column: 1 / span 5;
```



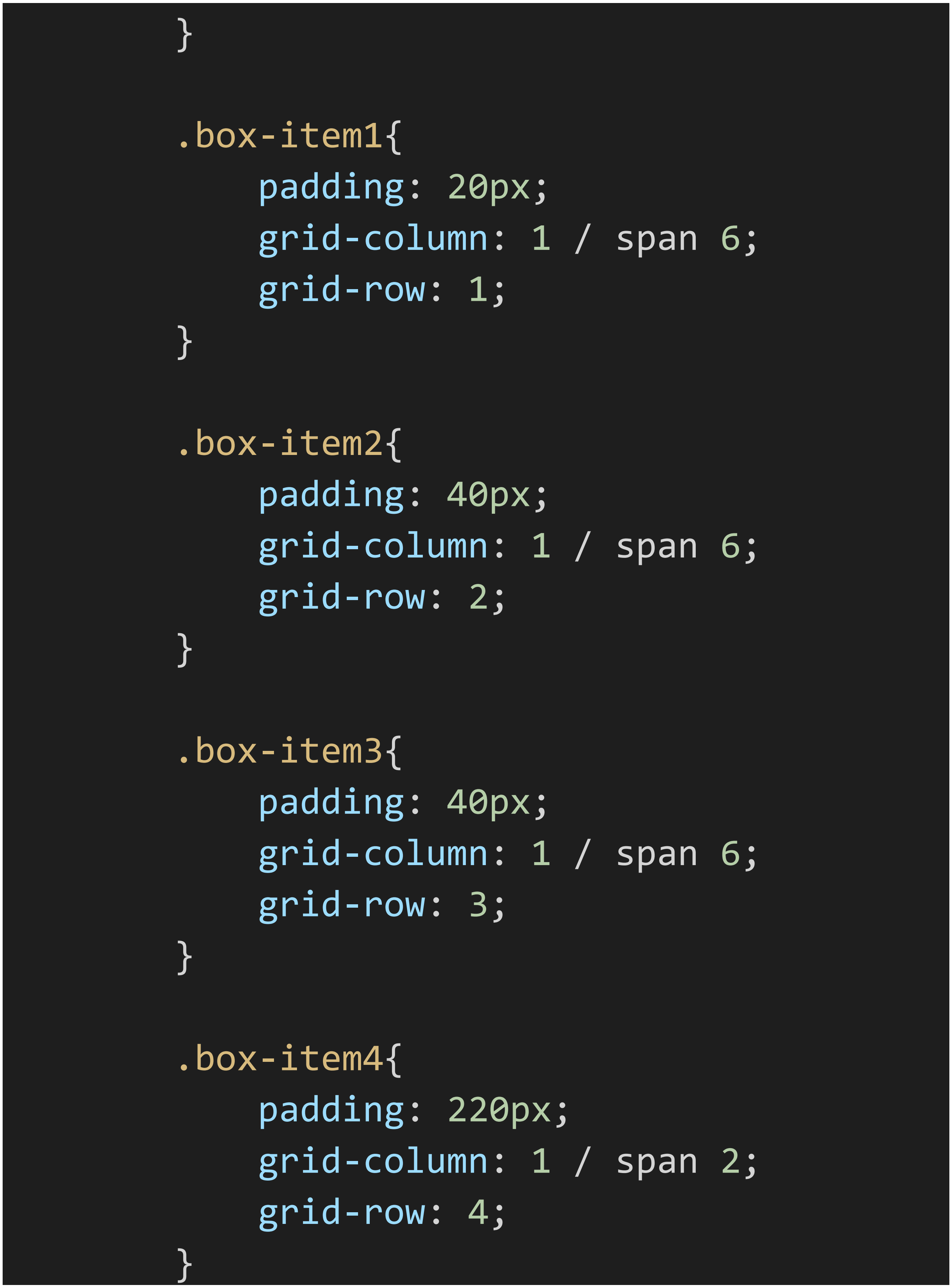
```
            grid-row: 1;
        }

        .item2{
            padding: 40px;
            grid-column: 1 / span 2;
            grid-row: 2 / span 3;
            background-color:rgb(227, 227,
212);
        }

        .item3{
            grid-column: 3 / span 3;
            grid-row: 2;
            padding: 40px;
        }

        .item5{
            grid-column: 3 / span 3;
            grid-row: 4;
            /* grid-column-start: 3;
            grid-column-end: 6; */
            padding: 170px;
        }
        .item6{
            grid-column: 1 / span 5;
            grid-row: 6;
            padding: 10px;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="grid-item
item1">Logo</div>
        <div class="grid-item item2">Side
bar Navigation</div>
        <div class="grid-item
item3">Header</div>
        <div class="grid-item item5">Body
Area</div>
        <div class="grid-item
item6">Footer</div>
    </div>
</body>
</html>
```

Layout 3

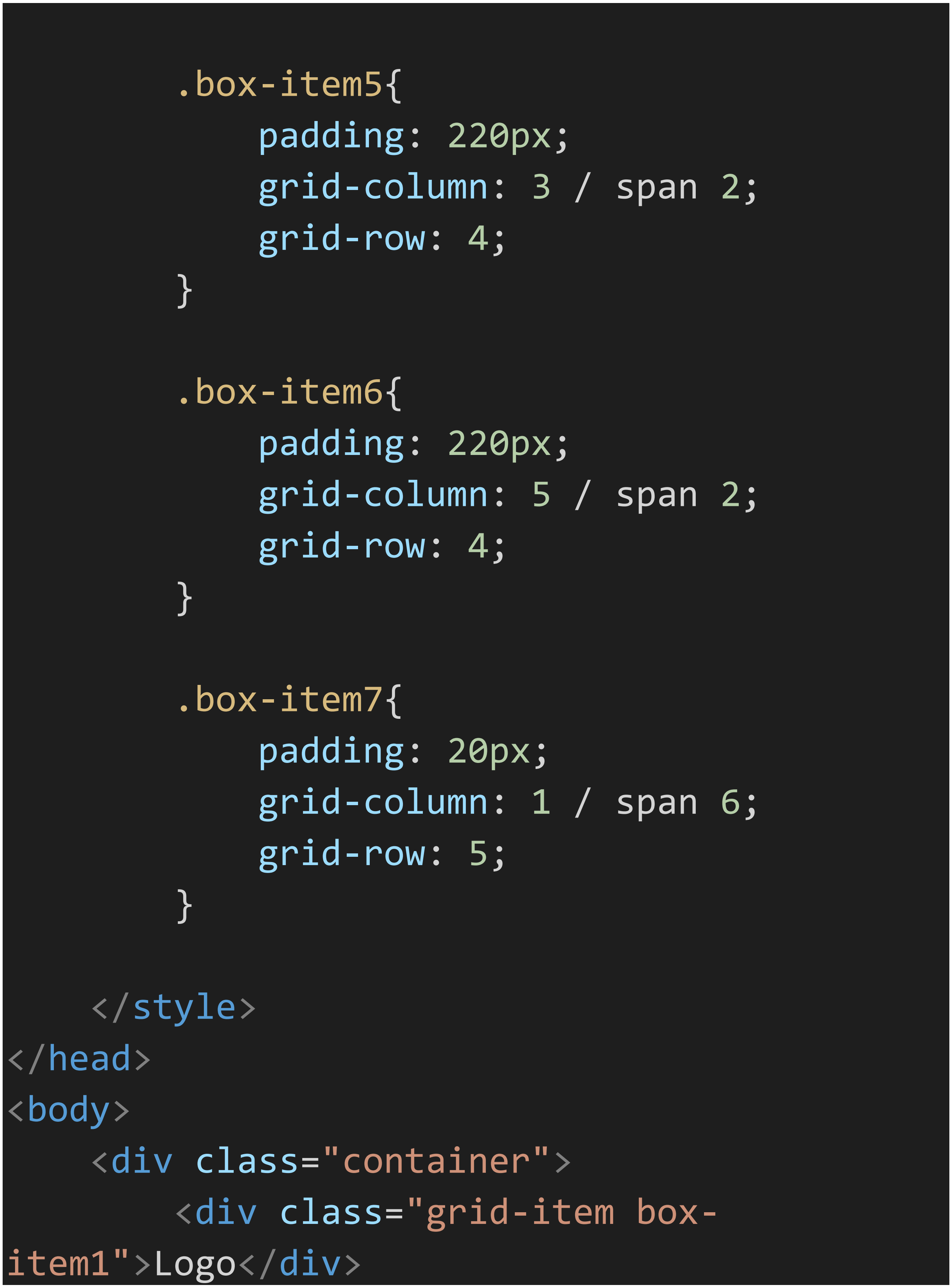
```
<!DOCTYPE html>
```



```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Grid Layout 3</title>

    <style>
         .container{
            display: grid;
            gap: 10px;
            background-color: lightgray;
            padding: 10px;
            padding-left: 200px;
            padding-right: 200px;
        }

        .grid-item{
            background-color: rgb(119, 113,
239);
            text-align: center;
            font-size: 20px;
```



```
        }

        .box-item1{
            padding: 20px;
            grid-column: 1 / span 6;
            grid-row: 1;
        }

        .box-item2{
            padding: 40px;
            grid-column: 1 / span 6;
            grid-row: 2;
        }

        .box-item3{
            padding: 40px;
            grid-column: 1 / span 6;
            grid-row: 3;
        }

        .box-item4{
            padding: 220px;
            grid-column: 1 / span 2;
            grid-row: 4;
        }
```

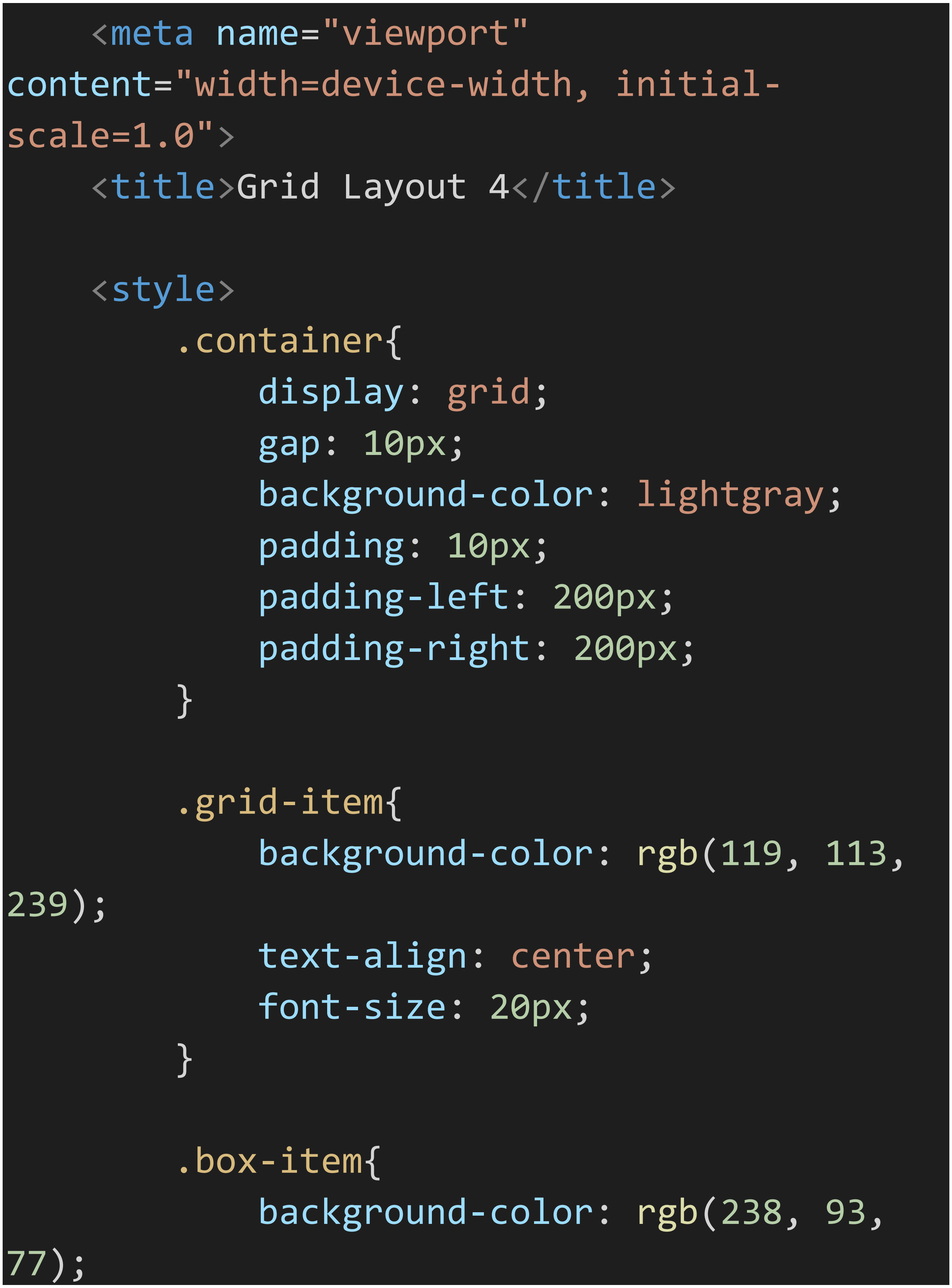


```
        .box-item5{
            padding: 220px;
            grid-column: 3 / span 2;
            grid-row: 4;
        }

        .box-item6{
            padding: 220px;
            grid-column: 5 / span 2;
            grid-row: 4;
        }

        .box-item7{
            padding: 20px;
            grid-column: 1 / span 6;
            grid-row: 5;
        }

    </style>
</head>
<body>
    <div class="container">
        <div class="grid-item box-
item1">Logo</div>
```

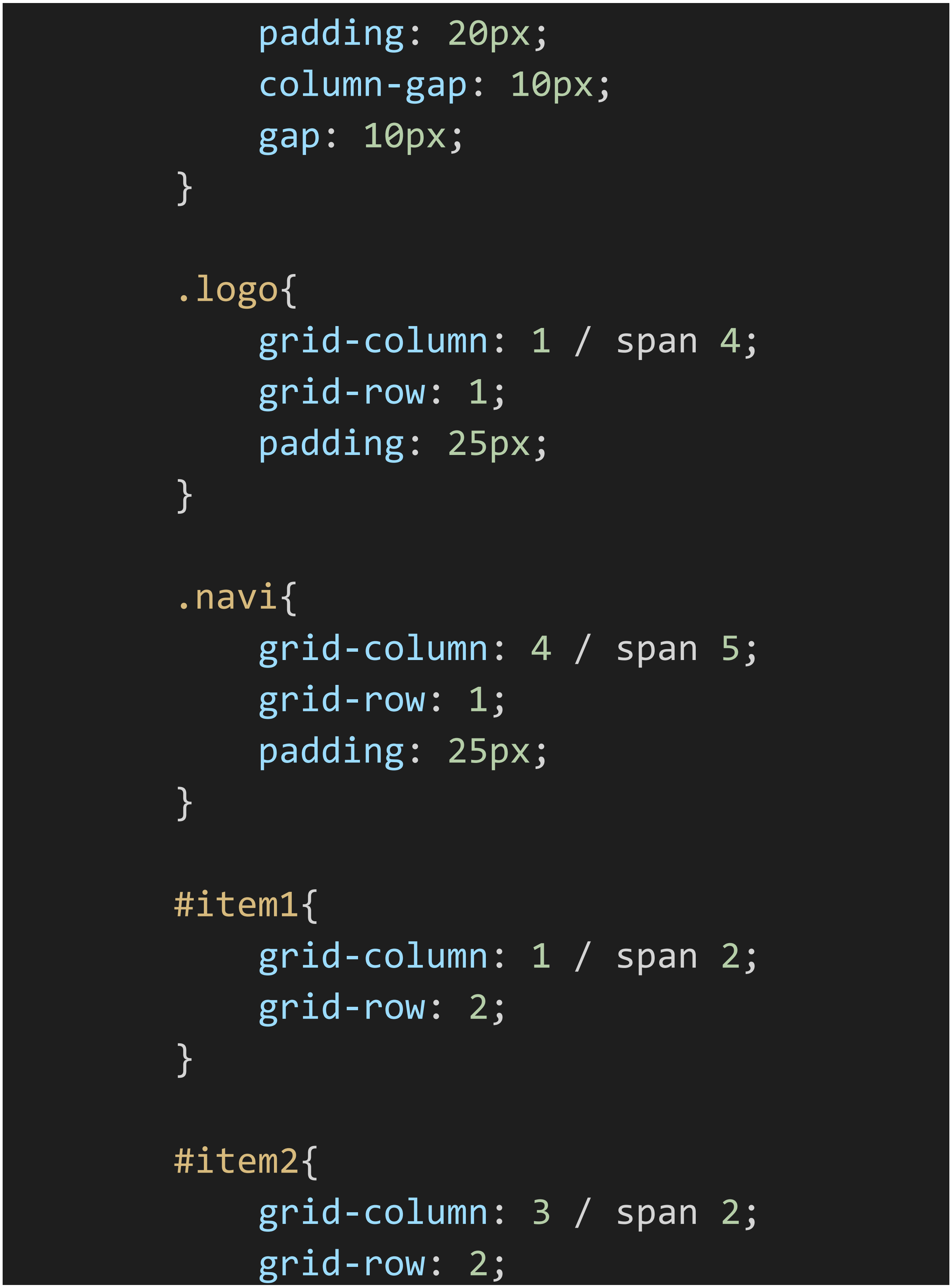
```
        <div class="grid-item box-
item2">Header</div>
        <div class="grid-item box-
item3">Text Area</div>
        <div class="grid-item box-item4">Box
1</div>
        <div class="grid-item box-item5">Box
2</div>
        <div class="grid-item box-item6">Box
3</div>
        <div class="grid-item box-
item7">Footer</div>
    </div>
</body>
</html>
```

Layout 4

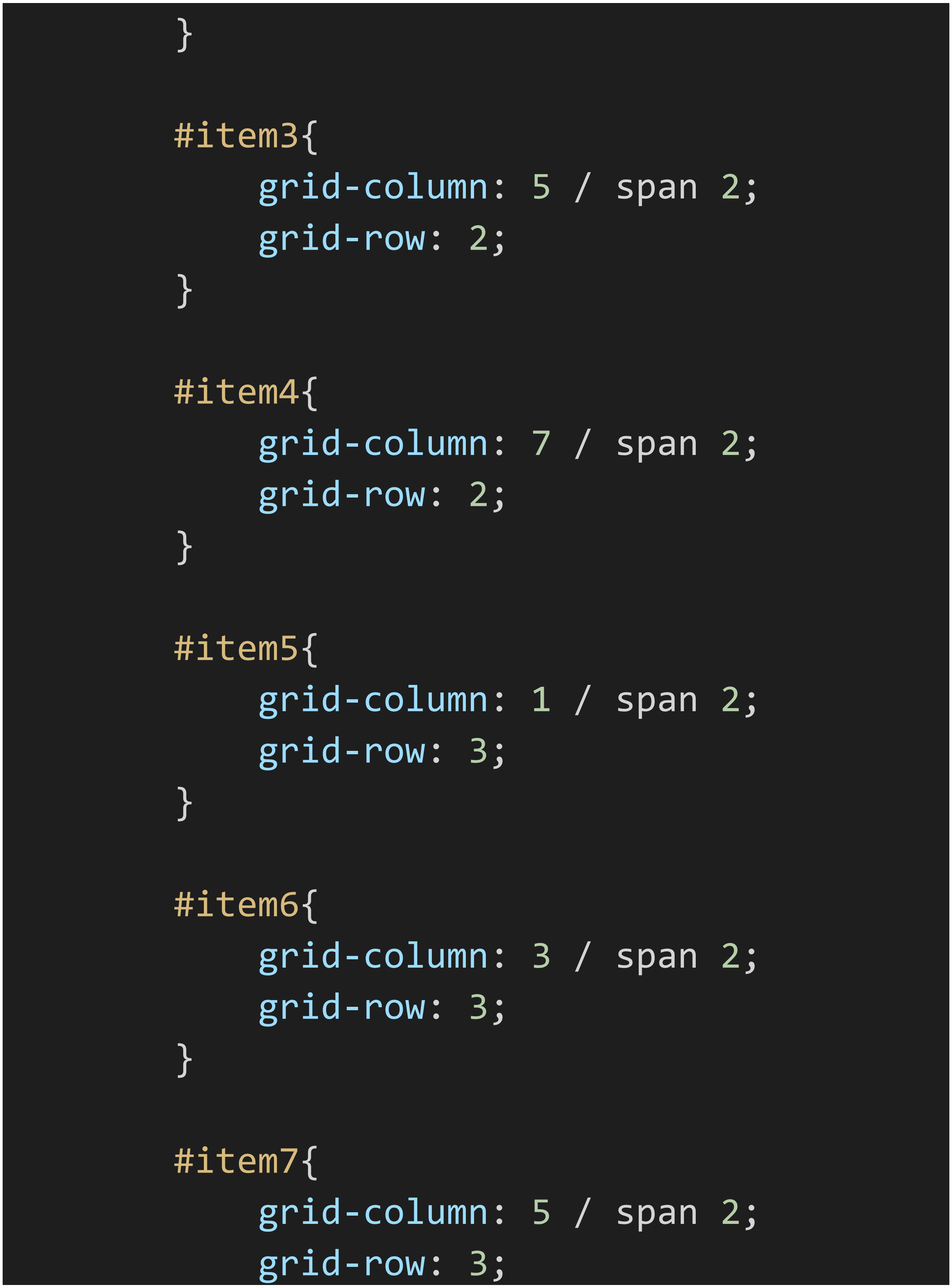
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
```



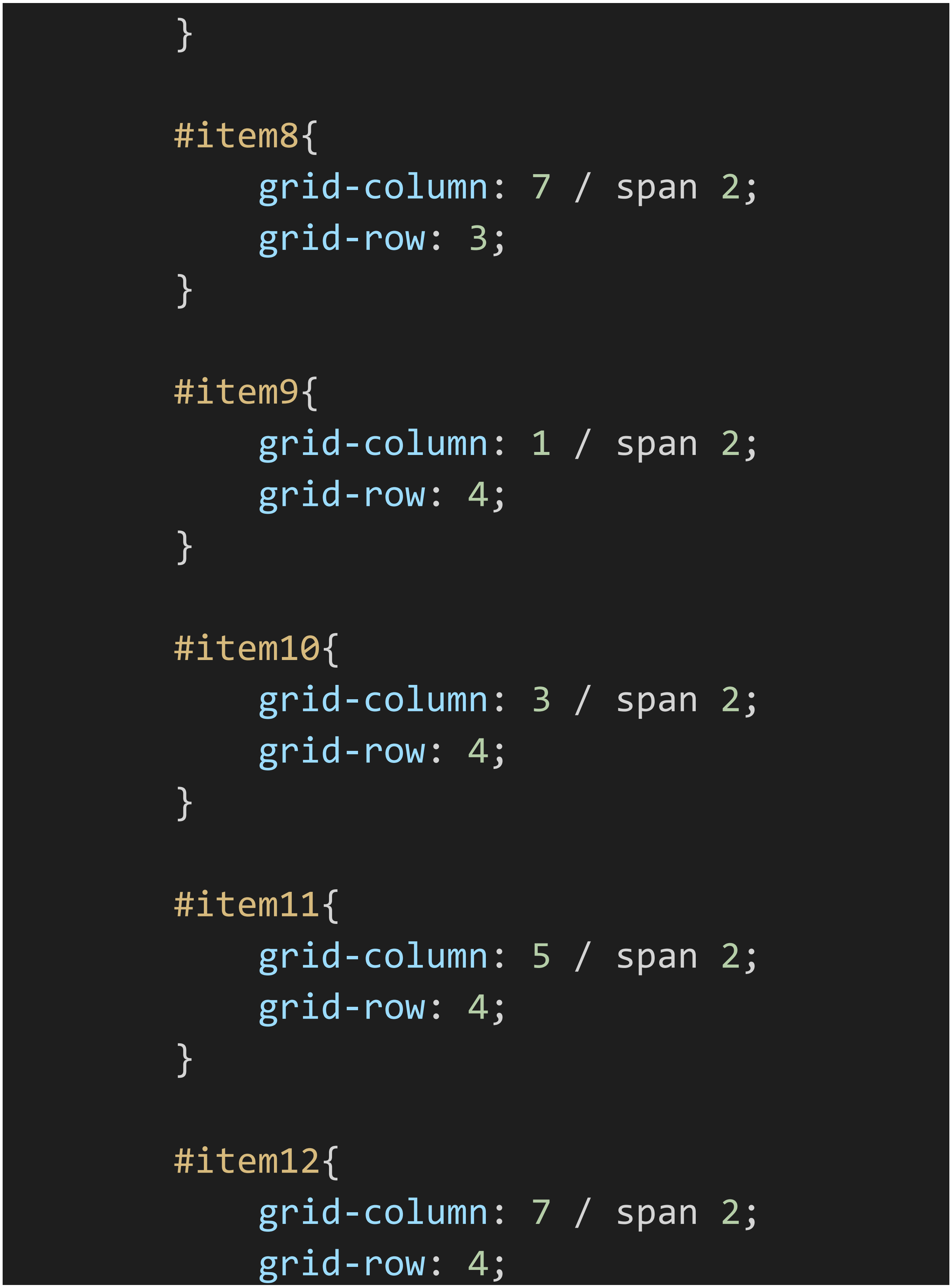
```
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <title>Grid Layout 4</title>

    <style>
        .container{
            display: grid;
            gap: 10px;
            background-color: lightgray;
            padding: 10px;
            padding-left: 200px;
            padding-right: 200px;
        }

        .grid-item{
            background-color: rgb(119, 113,
239);
            text-align: center;
            font-size: 20px;
        }

        .box-item{
            background-color: rgb(238, 93,
77);
```



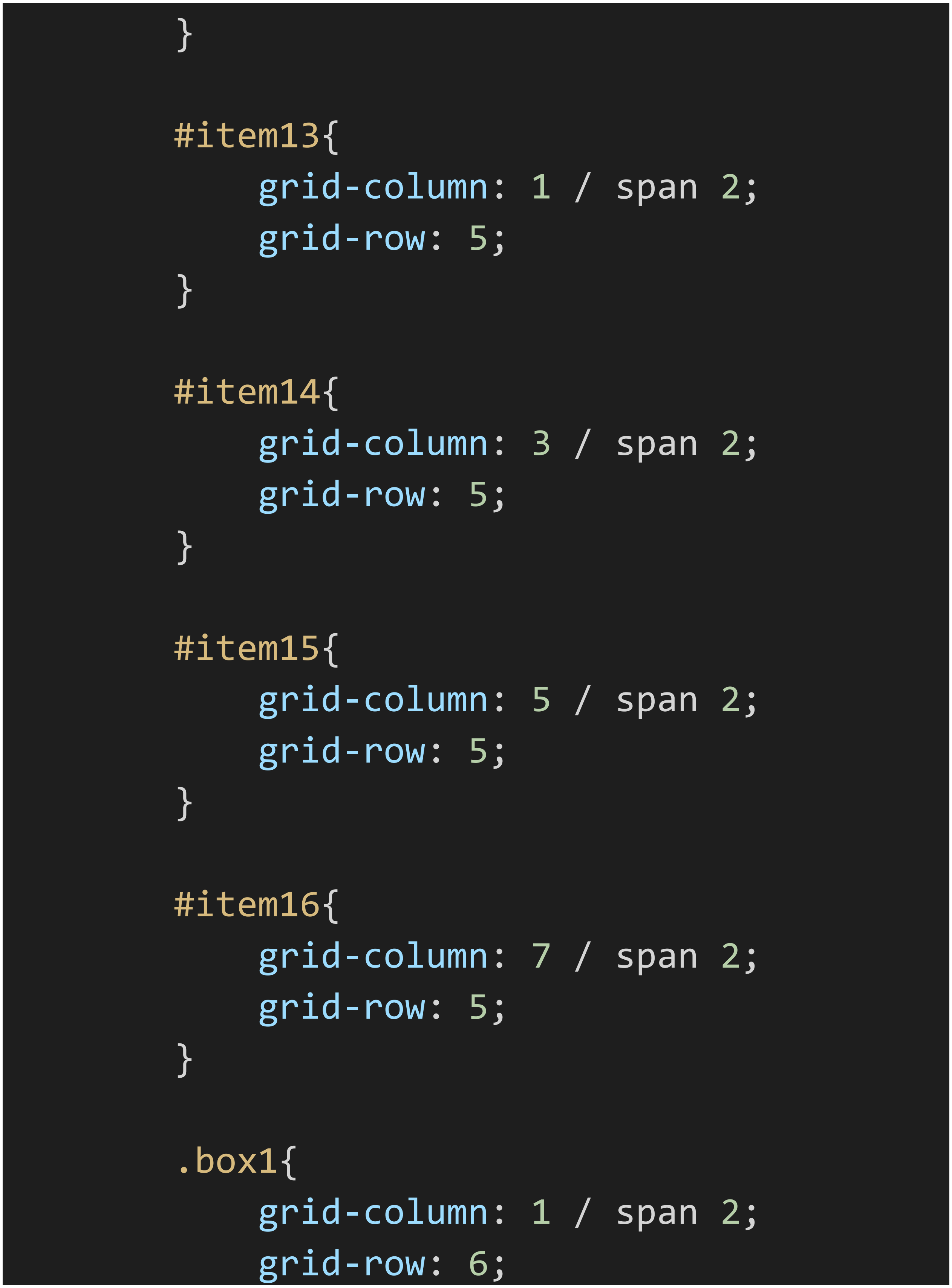
```
            padding: 20px;
            column-gap: 10px;
            gap: 10px;
        }

        .logo{
            grid-column: 1 / span 4;
            grid-row: 1;
            padding: 25px;
        }

        .navi{
            grid-column: 4 / span 5;
            grid-row: 1;
            padding: 25px;
        }

        #item1{
            grid-column: 1 / span 2;
            grid-row: 2;
        }

        #item2{
            grid-column: 3 / span 2;
            grid-row: 2;
```



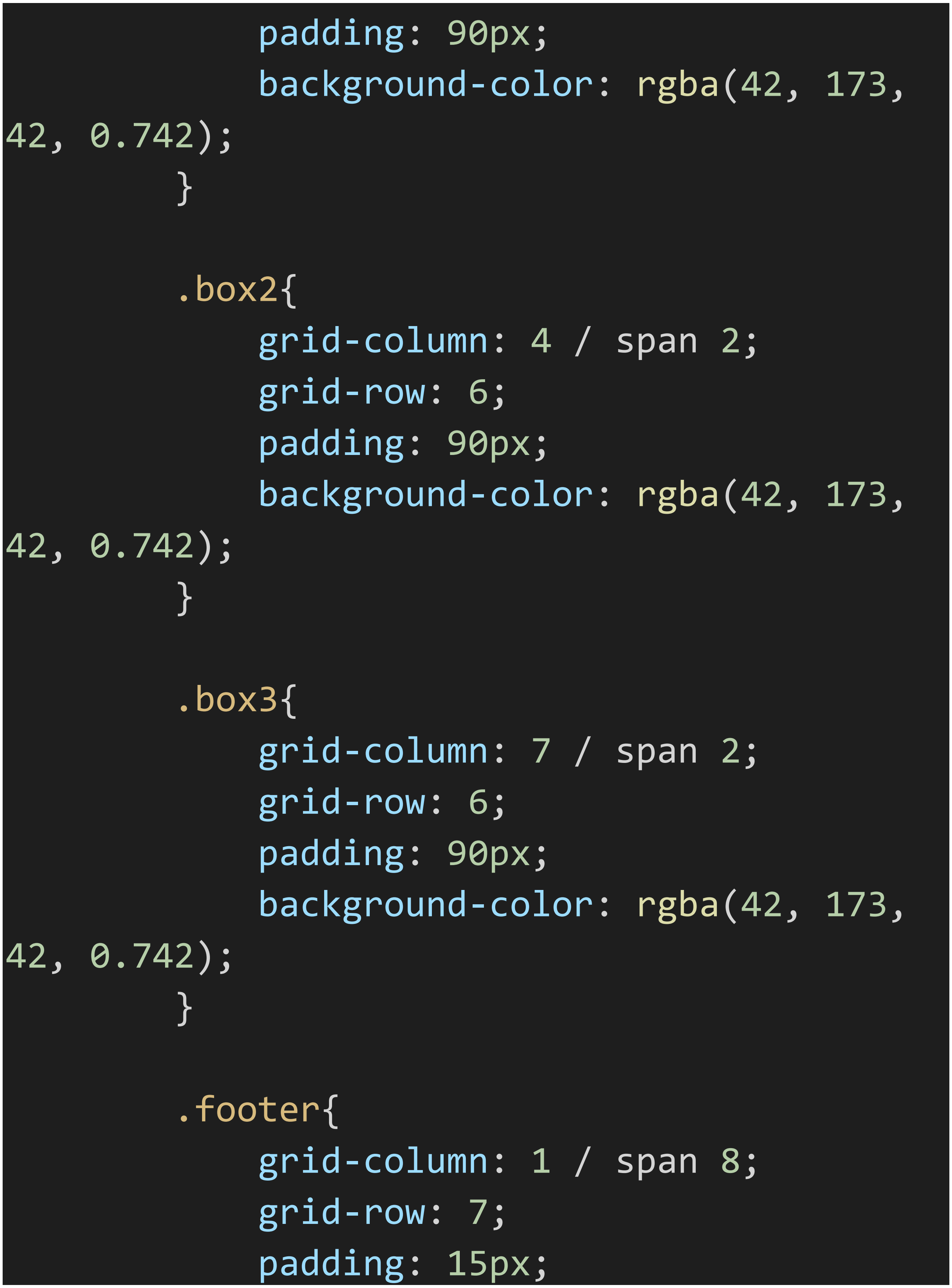
```
        }

        #item3{
            grid-column: 5 / span 2;
            grid-row: 2;
        }

        #item4{
            grid-column: 7 / span 2;
            grid-row: 2;
        }

        #item5{
            grid-column: 1 / span 2;
            grid-row: 3;
        }

        #item6{
            grid-column: 3 / span 2;
            grid-row: 3;
        }

        #item7{
            grid-column: 5 / span 2;
            grid-row: 3;
```



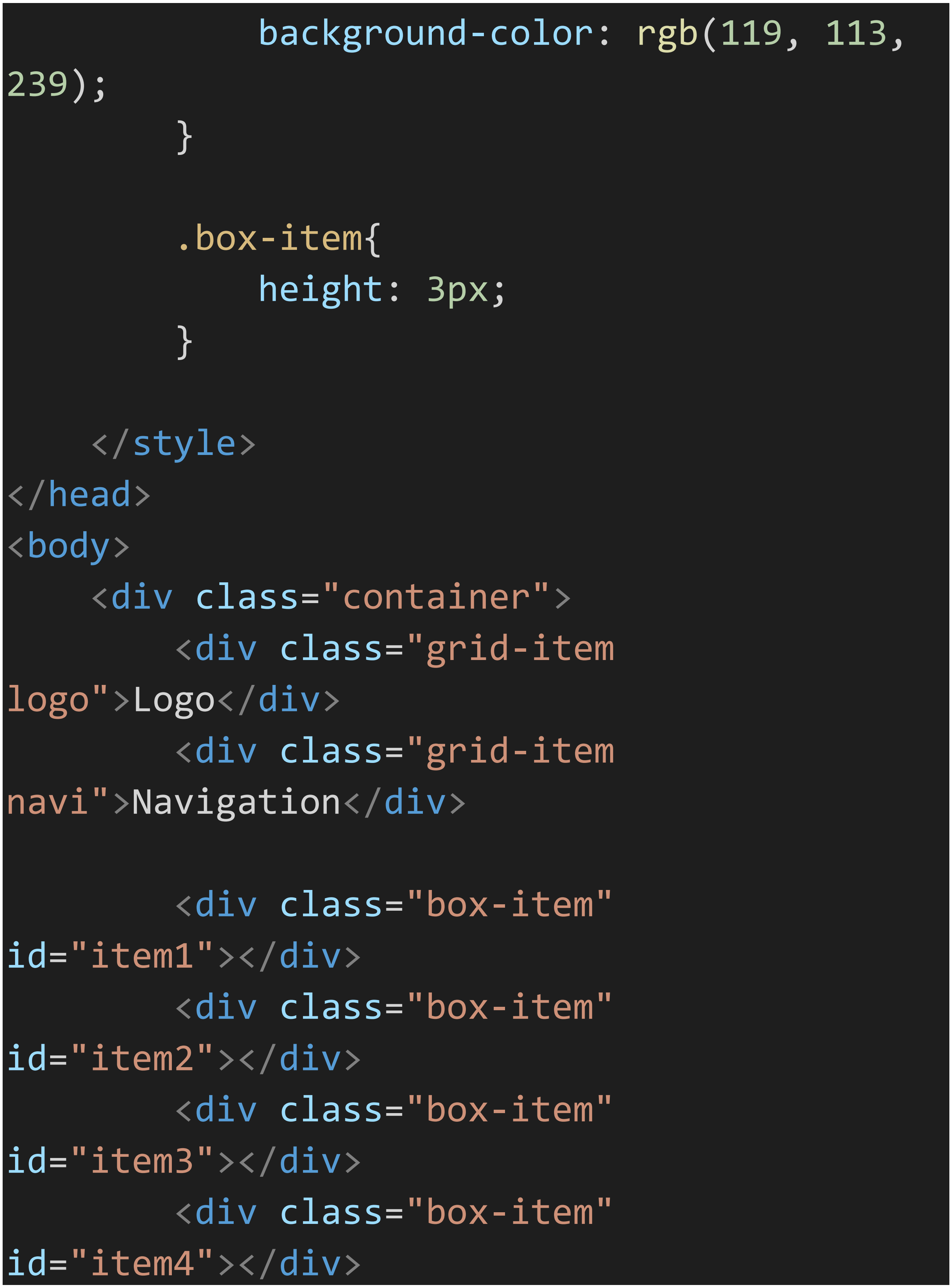
```
        }

        #item8{
            grid-column: 7 / span 2;
            grid-row: 3;
        }

        #item9{
            grid-column: 1 / span 2;
            grid-row: 4;
        }

        #item10{
            grid-column: 3 / span 2;
            grid-row: 4;
        }

        #item11{
            grid-column: 5 / span 2;
            grid-row: 4;
        }

        #item12{
            grid-column: 7 / span 2;
            grid-row: 4;
```



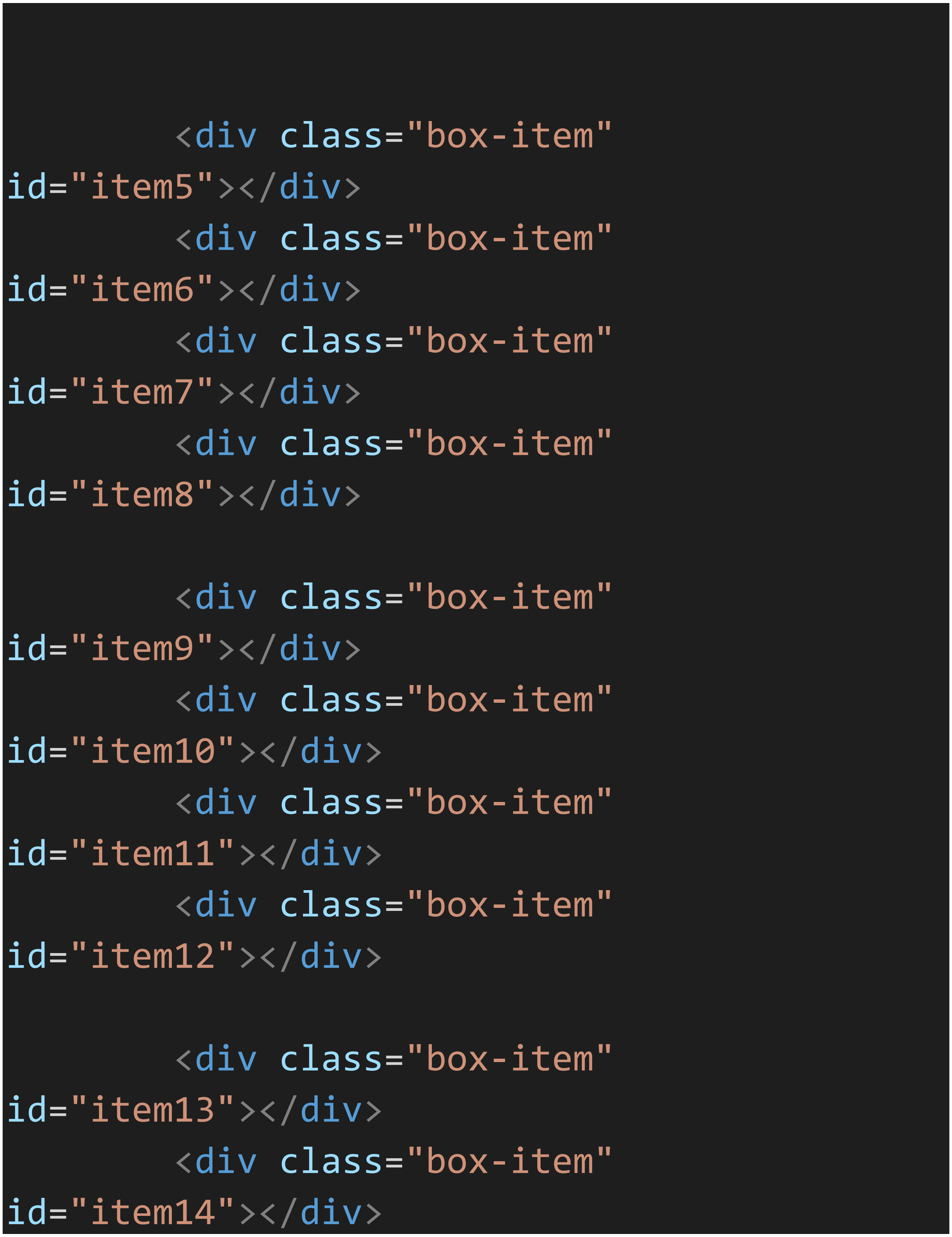
```
        }

        #item13{
            grid-column: 1 / span 2;
            grid-row: 5;
        }

        #item14{
            grid-column: 3 / span 2;
            grid-row: 5;
        }

        #item15{
            grid-column: 5 / span 2;
            grid-row: 5;
        }

        #item16{
            grid-column: 7 / span 2;
            grid-row: 5;
        }

        .box1{
            grid-column: 1 / span 2;
            grid-row: 6;
```



```
            padding: 90px;
            background-color: rgba(42, 173,
42, 0.742);
        }

        .box2{
            grid-column: 4 / span 2;
            grid-row: 6;
            padding: 90px;
            background-color: rgba(42, 173,
42, 0.742);
        }

        .box3{
            grid-column: 7 / span 2;
            grid-row: 6;
            padding: 90px;
            background-color: rgba(42, 173,
42, 0.742);
        }

        .footer{
            grid-column: 1 / span 8;
            grid-row: 7;
            padding: 15px;
```



```
            background-color: rgb(119, 113,
239);
        }

        .box-item{
            height: 3px;
        }

    </style>
</head>
<body>
    <div class="container">
        <div class="grid-item
logo">Logo</div>
        <div class="grid-item
navi">Navigation</div>

        <div class="box-item"
id="item1"></div>
        <div class="box-item"
id="item2"></div>
        <div class="box-item"
id="item3"></div>
        <div class="box-item"
id="item4"></div>
```



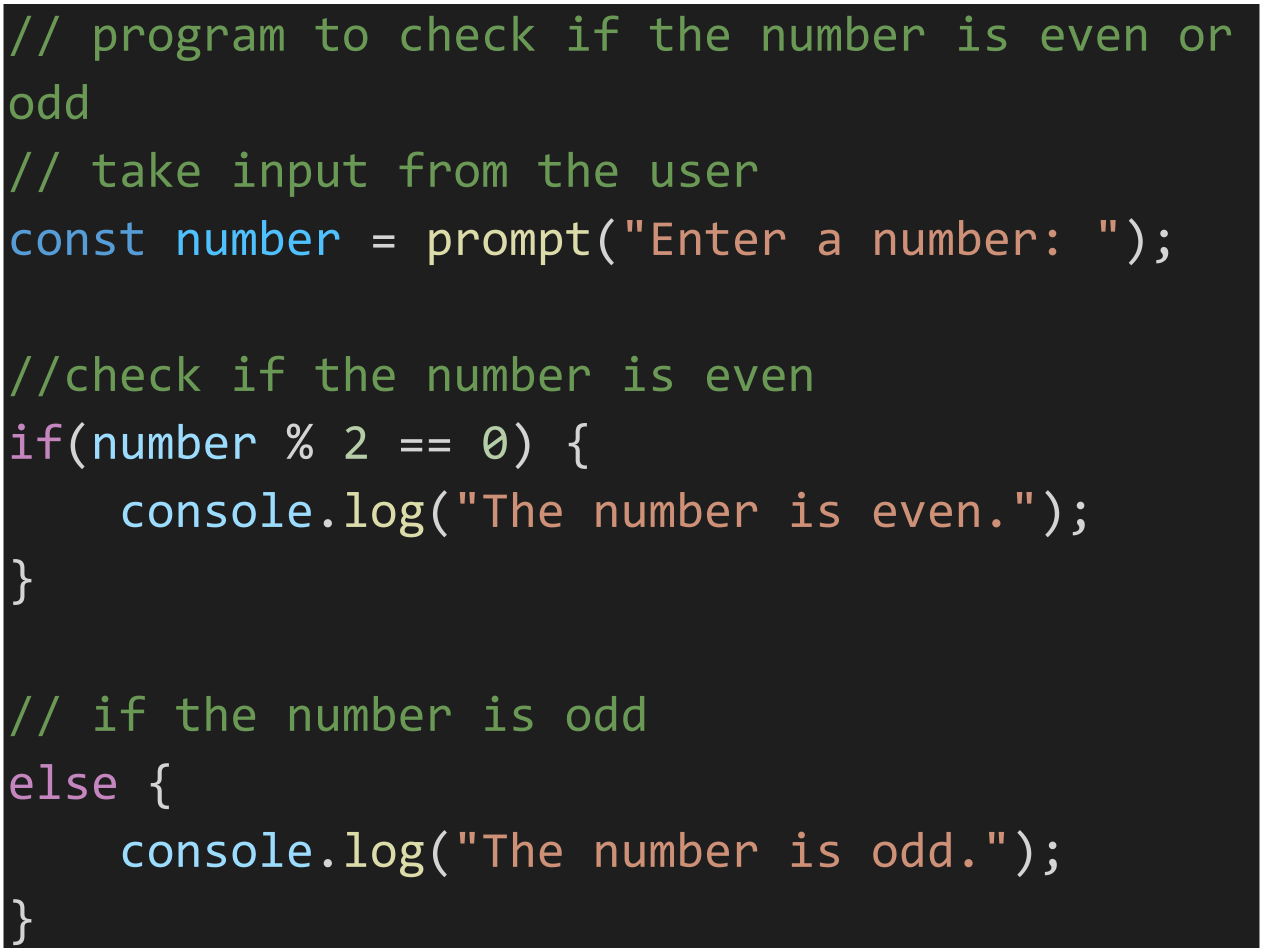
```
        <div class="box-item"
id="item5"></div>
        <div class="box-item"
id="item6"></div>
        <div class="box-item"
id="item7"></div>
        <div class="box-item"
id="item8"></div>

        <div class="box-item"
id="item9"></div>
        <div class="box-item"
id="item10"></div>
        <div class="box-item"
id="item11"></div>
        <div class="box-item"
id="item12"></div>

        <div class="box-item"
id="item13"></div>
        <div class="box-item"
id="item14"></div>
```

```
        <div class="box-item"
id="item15"></div>
        <div class="box-item"
id="item16"></div>

        <div class="grid-item box1">Box
1</div>
        <div class="grid-item box2">Box
2</div>
        <div class="grid-item box3">Box
3</div>
        <div class="grid-item
footer">Footer</div>
    </div>
</body>
</html>
```

Practical :- 25

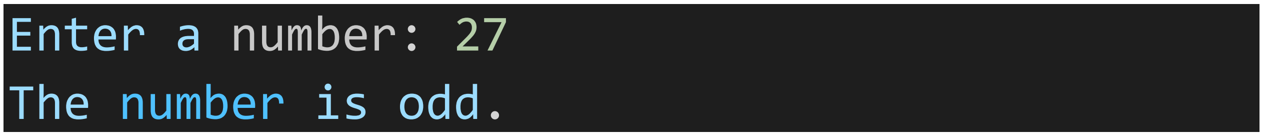
AIM :- Write a JavaScript to check if the number is even or odd.

Code :-



```
// program to check if the number is even or
odd
// take input from the user
const number = prompt("Enter a number: ");

//check if the number is even
if(number % 2 == 0) {
    console.log("The number is even.");
}

// if the number is odd
else {
    console.log("The number is odd.");
}
```

Output :-



```
Enter a number: 27
The number is odd.
```