

Experiment:2

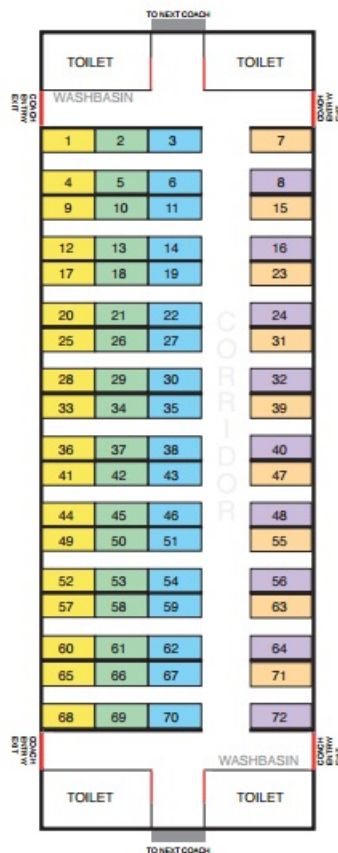
ARRAYS AND LOOPS

(Basic operations and application of logical operators)

1. Take a Sleeper class seat number as input and return a string that describes the berth position. For reference, you can see the below figure.

**COACH DIAGRAM FOR
 SLEEPER CLASS NON-AC
 COACH (ICF)**

LEGEND	
	Door
	Coach wall
■	Lower berth
■	Middle berth
■	Upper berth
■	Side Lower berth
■	Side Upper berth
■	Vestibule



Solution:

```

1 seat_number=int(input())
2 if 0<seat_number<=72:
3     if seat_number%8 == 1 or seat_number%8 ==4:

```

```

4         print("LB")
5     elif seat_number%8 == 2 or seat_number%8 == 5:
6         print("MB")
7     elif seat_number%8 ==3 or seat_number%8 ==6:
8         print("UB")
9     elif seat_number%8 ==7:
10        print("SL")
11    elif seat_number%8 ==0:
12        print("SU")
13 else:
14    print("invalid input")

```

2. Write a program to check whether a given positive integer is a perfect square or not.

Solution:

```

1 import math
2 x=math.sqrt(25)
3 int(x)
4 d=x-int(x)
5 if d!=0:
6     print("Not perfect square")
7 else:
8     print("square")

```

3. Write a program to generate the Fibonacci sequence up to a specified term "n".

Solution:

```

1 terms=int(input())
2 a=0
3 b=1
4 print(a)
5 print(b)
6 i=3
7 while i<=terms:
8     i+=1
9     print(a+b)
10    a,b=b,a+b

```

4. Write a program to check whether a given positive integer is a prime number or not.

Solution:

```
1 n=int(input())
2 counter=0;
3 for i in range(2,n//2):
4     if n%i==0:
5         counter=1;
6         break
7 if counter==1:
8     print("number is not prime")
9 else:
10    print("number is prime")
```