In [1]:

```
# 1) Write a function to check whether a number falls in a given range
def the_range(n):
    if n in range(2,8):
        print(" %s is in the given range"%str(n))
    else :
        print( " %s is outside the given range."%str(n))
the_range(9)
```

```
 9 is outside the given range.
```

In [5]:

```
# 2) Reducing the number of cards holding by half, rounded down

cards1=int(input("Enter number of cards in your hand :"))

number1 = (2)
result=cards1//number1
print(result,",is the result")
```

```
3 ,is the result
```

In [6]:

```
# 3) Print 'A' as many times as the generated number
import random

inputnumber1=int(input("Enter a positive number:"))
inputnumber2=inputnumber1+(10)
inputnumber3= random.randint(inputnumber1, inputnumber2)
print("Generated number is :",inputnumber3)
print("A "*inputnumber3)
```

```
Generated number is : 14
A A A A A A A A A A A A A A
```

In [7]:

```
# 4) Billing program
hrs1= float((input("Enter starting time in 24 hour time :")))
minhrs = 1
maxhrs = 2

if hrs1 < minhrs:
    print(float(input("Enter a valid number between 1 - 23 :")))
else :
    hrs2= float(input("Enter ending time :"))
    if hrs2 < maxhrs:
        print(float(input("Enter a valid time between 1 - 23 :")))
    else :
        total_hours= float(hrs2 - hrs1)

cost_per_hour = 5.50
total_user_bill = float(total_hours * cost_per_hour)
print("your total bill is :  $", total_user_bill,)
```

```
your total bill is :  $ 11.0
```

In [8]:

```
count = 0
import random
for n in range(10000):

    dice1 = random.randint(1,6)
    dice2 = random.randint(1,6)
    if dice1==dice2:
```

```
        count +=1
percent=(count/10000)*100
print(f"The persentage of getting doubles is : {percent}%")
```

```
The persentage of getting doubles is : 16.77%
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```
def
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```