## **PRACTICE SHEET SOLUTION**

Q1) b) 3

Q2)

secret = 1234

attempts = 0

while attempts < 3:

guess = int(input("Enter the password: "))

if guess == secret:

print("Unlocked")

break

else:

print("Try again")

attempts += 1

Q3)

i)

n = int(input())

for i in range(n):

print(4 + 3 \* i)

ii)

n = int(input())

print([3 \* (4 \*\* i) for i in range(n)])

Q4)

n = int(input("Enter number of passengers: "))

if n == 1:

probability = 1.0

else:

probability = 0.5

print(f"The probability that the {n}th person gets their own seat is: {probability}")

Q5)

n = int(input("Enter the number of bulbs: "))

count = 0

for i in range(1, n + 1):

root = int(i \*\* 0.5)

if root \* root == i:

count += 1

print(f"Number of bulbs that are on after {n} rounds: {count}")

Q6)

def generate\_password(name, year, color):

    part1 = name[0].upper()

    part2 = str(year)[::-1]

    part3 = color[:2].upper()

    password = part1 + part2 + part3

    print("Your funny password is:", password)

# Example usage

generate\_password("khushbu", 2003, "purple")

Q7)

def simple\_interpreter(command):

try:

exec(command)

except Exception as e:

print("Error:", e)

simple\_interpreter("print(5 + 10)")

simple\_interpreter("for i in range(3): print(i)")

Q8)

import math

u, v, a, s = map(int, input().split())

# Calculate final velocity squared

final\_velocity\_squared = u \* u - 2 \* a \* s

# If negative, set it to 0 (can't have negative velocity squared)

if final\_velocity\_squared < 0:

final\_velocity\_squared = 0

# Calculate final velocity

final\_velocity = math.sqrt(final\_velocity\_squared)

if final\_velocity > v:

print("No")

else:

print("Yes")

Q9)

n = int(input())

sum\_n = n \* (n + 1) // 2 # Integer division

if sum\_n % 2 == 0:

print(n)

else:

print(n - 1)

Q10)

a, b, c, d, k = map(int, input().split())

count = abs(c - a) + abs(d - b)

if k >= count and (k - count) % 2 == 0:

print("YES")

else:

print("NO")

Q11)

n = int(input())

# Upper part of the pattern

for i in range(1, n + 1):

print('\*' \* i)

# Lower part of the pattern

for i in range(n - 1, 0, -1):

print('\*' \* i)

Q12)

n = int(input())

# First line: print all characters from A to (2n-1)th character

for i in range(2 \* n - 1):

print(chr(ord('A') + i), end=' ')

print() # New line

for i in range(1, n):

# Left part: characters from A to (n-i)th character

for j in range(n - i):

print(chr(ord('A') + j), end=' ')

# Middle spaces

for j in range(2 \* i - 1):

print(' ', end=' ')

# Right part: characters from (n+i)th to end

for j in range(n - i):

print(chr(ord('A') + n + i + j - 1), end=' ')

print() # New line

Q13)

reactant = 100 # initial amount in grams

hours = 0

while reactant > 10: # reaction completes when only 10 grams are left

reactant -= 0.1 \* reactant # use 10% of current amount

hours += 1

print("Reaction completed in", hours, "hours")

Q14)

n = int(input())

if n == 1:

print("3")

else:

# Construct number: '3' + (n-2 zeros) + '3'

print("3" + "0" \* (n - 2) + "3")

Q15)

n = int(input("Enter a number: "))

str\_num = str(n)

total = 0

for i in range(len(str\_num)):

total += int(str\_num[i]) \*\* (i + 1)

if total == n:

print(f"{n} is a Disarium number.")

else:

print(f"{n} is not a Disarium number.")