09- Functions

Ex. No. : 9.1 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

.

complete function to implement coin change making problem i.e. finding the minimum number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

return the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value

Program:

```
def coinChange(amount):
    coins=[1,2,3,4]
    dp = [float('inf')] * (amount + 1)
    dp[0] = 0
```


```
for i in range(1, amount + 1):
    for coin in coins:
        if i - coin >= 0:
            dp[i] = min(dp[i], dp[i - coin] + 1)
return dp[amount] if dp[amount] != float('inf') else -1
```

Ex. No. : 9.2 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Check Product of Digits

Write a code to check whether product of digits at even places is divisible by sum of digits at odd place of a positive integer.
Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

For example:

Test	Result
print(productDigits(1256))	True

Test	Result
print(productDigits(1595))	False

Program:

```
def productDigits(n):
    eve=[]
    odd=[]
    count=0
    while(n>0):
        r=n%10
        if(count%2==0):
            eve.append(r)
        else:
            odd.append(r)
        n//=10
        count+=1
    prod=1
    for i in eve:
        prod*=i
    if(prod%(sum(odd))==0):
        return True
    else:
        return False
```

	Test	Expected	Got	
✔	print(productDigits(1256))	True	True	✔
✔	print(productDigits(1595))	False	False	✔

\Ex. No. : 9.3 **Date: 01.06.24**

Register No.: 230701385 **Name: S. Vishwak**

.

Given a number with maximum of 100 digits as input, find the difference between the sum of odd and even position digits.

Input Format:

Take a number in the form of String from stdin.

Output Format:

Print the difference between sum of even and odd digits

Example input:

1453

Output:

1

Explanation:

Here, sum of even digits is 4 + 3 = 7

sum of odd digits is 1 + 5 = 6.

Difference is 1.

Note that we are always taking absolute difference

Program:

```
def differenceSum(n):
    eve=[]
    odd=[]
    count=0
    while(n>0):
```

```
    r=n%10
    if(count%2==0):
        eve.append(r)
    else:
        odd.append(r)
    n//=10
    count+=1
return abs(sum(odd)-sum(eve))
```

	Test	Expected	Got	
✔	`print(abundant(12))`	Yes	Yes	✔
✔	`print(abundant(13))`	No	No	✔

Ex. No. : 9.4 **Date: 01.06.24**

Register No.: 230701385 **Name S. Vishwak**

.

Ugly number

A number is considered to be ugly if its only prime factors are 2, 3 or 5.
[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, …] is the sequence of ugly numbers.
Task:
complete the function which takes a number n as input and checks if it's an ugly number.
return ugly if it is ugly, else return not ugly
Hint:
An ugly number U can be expressed as: U = 2^a * 3^b * 5^c, where a, b and c are nonnegative integers.

For example:

Test	Result
print(checkUgly(6))	ugly
print(checkUgly(21))	not ugly

Program:

```
def checkUgly(n):
    def maxDivide(a, b):
        while a % b == 0:
            a = a / b
        return a
    n = maxDivide(n, 2)
```

```
n = maxDivide(n, 3)
n = maxDivide(n, 5)
if(n==1):
    return("ugly")
else:
    return("not ugly")
```

	Test	Expected	Got	
✔	`print(checkUgly(6))`	`ugly`	`ugly`	✔
✔	`print(checkUgly(21))`	`not ugly`	`not ugly`	✔

Ex. No. : 9.5 **Date: 01.06.24**

Register No.: 230701385 **Name S. Vishwak**

.

Automorphic number or not

An automorphic number is a number whose square ends with the number itself. For example, 5 is an automorphic number because 5*5 =25. The last digit is 5 which same as the given number.

If the number is not valid, it should display “Invalid input”.
If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:
Take a Integer from Stdin
Output Format:
Print Automorphic if given number is Automorphic number, otherwise Not Automorphic
Example input: 5 Output: Automorphic Example input: 25 Output: Automorphic Example input: 7 Output: Not Automorphic

For example:

Test	Result
print(automorphic(5))	Automorphic

Program:

```
def automorphic(n):
    sq=n*n
    if (n % 10 != sq % 10) :
        return ("Not Automorphic")
    return ("Automorphic")
```

	Test	Expected	Got	
✔	print(automorphic(5))	Automorphic	Automorphic	✔
✔	print(automorphic(7))	Not Automorphic	Not Automorphic	✔