Call sleep\_time()

Call wake\_time()

If cycles\_slept>=4 or 5

elif cycles\_slept>0 and <4 or 5

If cycles\_slept>=4 or 5

if cycles\_slept>0and<4 or 5

Print msg

Call waking\_time(4)

Print msg

Call Sleeping\_time(5)

Stop

Start

If choice\_1 = 2

If choice\_1=1

Input the choice\_1, choice\_2 and cycles\_slept

From datetime import datetime and import time

Stop

Use time.strptime() time.mktime()

to display sleeping\_time in time format

Subtract sleep\_sec from input time(time\_to\_wake\_up)

sleep\_sec = (N-cycles\_slept)\*5400 + (N-cycles\_slept)\*900

sleeping\_time(N)

Stop

Use time.strftime() and time.localtime() to display the waking\_time

Add sleep\_sec and current\_time and store it in waking\_time

waking\_time(n)

sleep\_sec = ((n-cycles\_slept)\*5400+(n-cycles\_slept)\*900)

Flowcharts for user-defined functions

Algorithm

Step 1 : Start

Step 2 : import datetime and time modules

Step 3 : Use date\_time module for the current time

Step 4 : Store the input in choice\_1 whether the user wants to sleep now or to wake up at specified time

Step 5 : Store the input in choice\_2 whether the user wants to go with 4 sleep cycles or 5 sleep cycles

Step 6 : Input from the user how many sleep cycles he has already completed and store it in cycles\_slept

Step 7 : Define waking\_time(n) function to print the appropriate wake time according to the input

7.1 Extra seconds that has to be slept store in sleep\_sec

sleep\_sec = ((n-cycles\_slept)\*5400+(n-cycles\_slept)\*900)

7.2 Use time.strftime() and time.localtime() to print the waking\_time

7.3 Make print statement of waking\_time by adding sleep\_sec to the current\_time

Step 8 : Define sleeping\_time(N) function to print the appropriate time to sleep according to the input

8.1 Extra seconds that has to be slept subtract from the current time and store in sleeping\_time

8.2 Use time.strptime() and time.mktime() to convert the input time into seconds

sleep\_sec = (N-cycles\_slept)\*5400 + (N-cycles\_slept)\*900

8.3 Make a print statement of sleeping\_time by subtracting seconds that has to sleep from time the user wants

to wake up

Step 9 : Define wake\_time() function to print the wake time

9.1 Take if elif else statements according to the choice\_2

9.2 if choice\_2 = 3 call waking\_time(4) function in wake\_time() function

9.3 elif choice\_2 = 4 call waking\_time(5) function in wake\_time() function

9.4 Call waking\_time() function in elif statement to print the wake time

Step 10 : Define sleep\_time() function to print the sleep time

10.1 Take if elif else statements according to the choice\_2

10.2 if choice\_2 = 3 call sleeping\_time(4) in sleep\_time() function

10.3 elif choice\_2 = 4 call sleeping\_time(5) in sleep\_time() function

10.4 Call sleeping\_time() function in elif statement to print the sleep time

Step 11 : Take if elif else statement according to choice\_1

11.1 Call wake\_time() function in if statement

11.2 Call sleep\_time() function in elif statement

11.3 print invalid input in else statement

Wake-up time Calculator

import time

from datetime import datetime # importing datetime module to get the current time

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Welcome to Wake-up time Calculator\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

now = datetime.now()

Current\_time = int(now.strftime("%H%M%S"))

now=time.time()

local\_time = time.ctime(now) *# input to ask whether the end user wants to sleep now or want to*

*wake up at specified time*

choice\_1 = input("\nEnter 1 to sleep now or 2 to wake up at specified time:\n")

choice\_2 = input("\nEnter 3 to go with four sleep cycles or enter 4 to go with 5 sleep cycles:\n") # no.of cycles slept today

cycles\_slept = int(input("How many cycles have you slept today?:\n"))

def waking\_time(N*): #function to print wake time*

sleep\_sec = (N-cycles\_slept)\*5400 + (N-cycles\_slept)\*900

waking\_time = time.localtime(now + sleep\_sec)

waking\_time = time.strftime("%H:%M:%S", waking\_time)

print("You should wake up at:",waking\_time)

def sleeping\_time(n): *#function to print sleep time*

time\_to\_wake\_up = input("At what time do you want to wake up?\n\nEnter time in the format of 08:45:30, Aug 24 2019\n")

time\_to\_wake\_up = time.strptime(time\_to\_wake\_up,"%H:%M:%S, %b %d %Y")

time\_to\_wake\_up = time.mktime(time\_to\_wake\_up)

sleep\_sec = ((n-cycles\_slept)\*5400+(n-cycles\_slept)\*900)

sleeping\_time = time.localtime(time\_to\_wake\_up - sleep\_sec)

sleeping\_time = time.strftime("%H:%M:%S, %b %d %Y",sleeping\_time)

print("you should sleep at:",sleeping\_time)

def wake\_time(): *# user-defined function to suggest the best time to wake up*

if choice\_2 == '3': *# condition for 4 sleep cycles*

if cycles\_slept >= 4:

print("You have slept enough no.of cycles.\n-\*-\*-\*-\*-\*-Have a nice day!!-\*-\*-\*-\*-\*-")

elif cycles\_slept >= 0 and cycles\_slept<4:

waking\_time(4)

elif choice\_2 == '4': *# condition for 5 sleep cycles*

if cycles\_slept >= 5:

print("You have slept enough no.of cycles.\n-\*-\*-\*-\*-\*-Have a nice day!!-\*-\*-\*-\*-\*-")

elif cycles\_slept >= 0 and cycles\_slept < 5:

waking\_time(5)

else:

print("Invalid input")

def sleep\_time(): *# user-defined function to tell the end user the appropriate sleeping time*

if choice\_2 == '3':

if cycles\_slept >= 4: *# condition for 4 sleep cycles*

print("You have slept enough no.of sleep cycles.\n-\*-\*-\*-\*-\*-Have a nice day!!-\*-\*-\*-\*-\*-")

elif cycles\_slept >= 0 and cycles\_slept <4 :

sleeping\_time(4)

elif choice\_2 == '4':

if cycles\_slept >= 5: *#condition for 5 sleep cycles*

print("You have slept enough no.of cycles.\n-\*-\*-\*-\*-\*-Have a nice day!!-\*-\*-\*-\*-\*-")

elif cycles\_slept >= 0 and cycles\_slept < 5 :

sleeping\_time(5)

else:

print("Invalid input")

if choice\_1 == '1':

wake\_time() *#calling wake\_time function*

elif choice\_1 == '2':

sleep\_time() *#calling sleep\_time function*

else:

print("Invalid input")

enter=input("Enter 1:")

if enter == '1':

print("\*\*\*\*\*\*\*\*Thank you\*\*\*\*\*\*\*\*")

else:

print("\*\*\*\*\*\*Good bye\*\*\*\*\*\*\*\*")