TREDMILL DASHBOARD

E/18/115 – GOWSIGAN A.

Introduction

In this project, we were given the task to build a program that gives useful insights to a treadmill user based on the conditions at which the treadmill is operated. The program should be able get the operational parameters in any unit and necessary unit conversions to do the required calculations should be included in the program. The project was divided into two major parts where 1st milestone get time as an input and calculate the outputs but in the 2nd milestone the outputs must be given in real time. So, our goal was to achieve these 2 milestones in the program we design.

1st milestone

In this part we created a program that give outputs using the inputs. Required inputs and derived outputs are listed below

Inputs that are required to run program,

- The rate at which the motor is rotating (RPM)
- The radius of the motor shaft
- Weight of the person
- Height of the person
- Time duration that person was walking/ running

The derived outputs are:

- The speed
- The distance walked/ ran on the treadmill
- Calories burnt on that time period
- Number of steps taken

2nd milestone

In the second part, our objective is to give user a real time outputs to the user when the program is started. The time is taken to the program and it will be displayed with varying outputs with the time.

Inputs that are required to run program,

- The rate at which the motor is rotating (RPM)
- The radius of the motor shaft
- Weight of the person
- Height of the person

These has to be given as inputs and when the user press start the below outputs will be displayed according to time until the user need to stop.

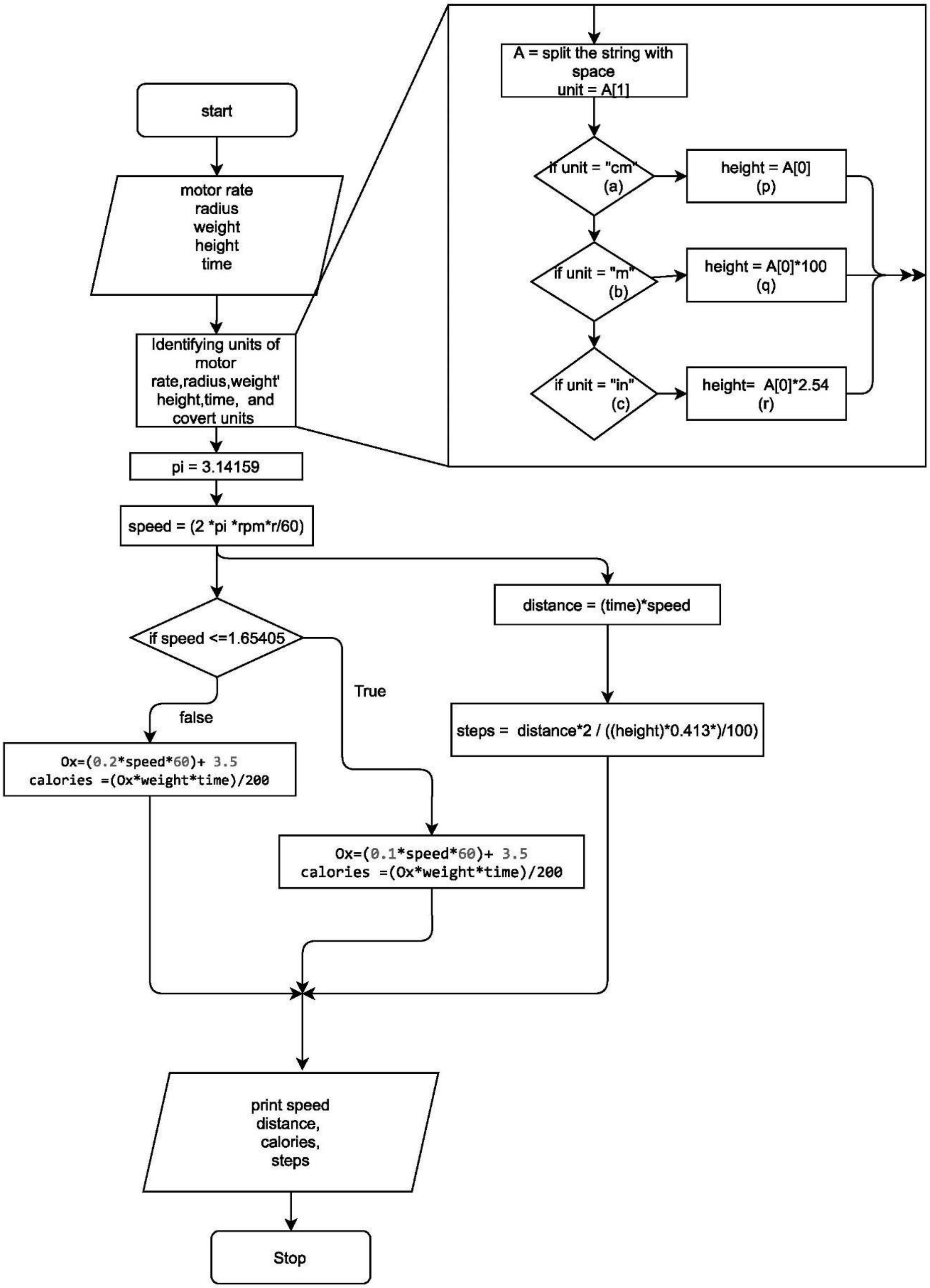
- The speed
- The distance walked/ ran on the treadmill
- Calories burnt on that time period
- Number of steps taken

Bonus milestone

We expect to develop a GUI (Graphical User Interface) which would make the use of this program very friendly to the user.

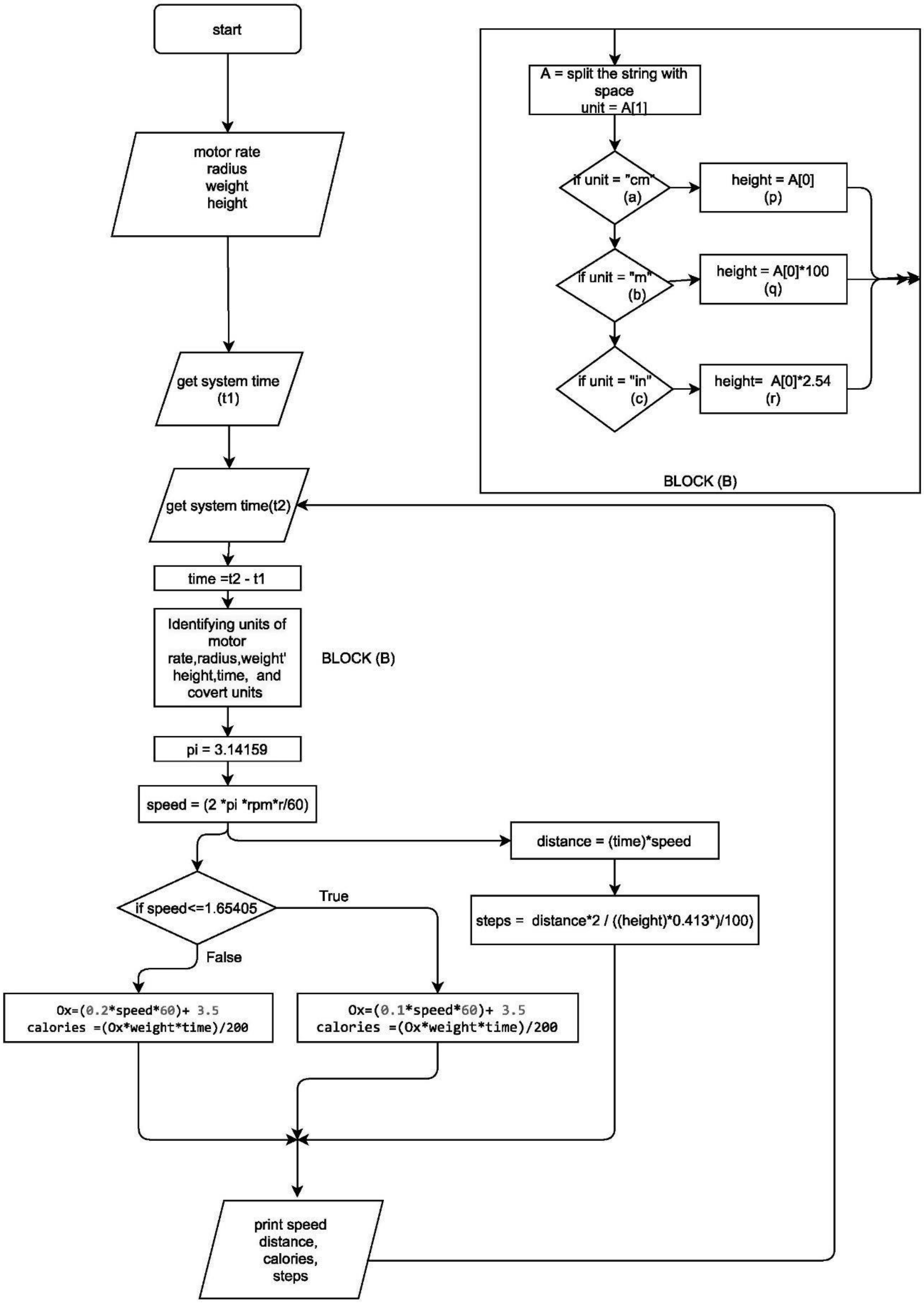
Flow chart (a)

The flow chart that we have designed for 1st milestone



Flow chart (b)

The flow chart that we have designed for 2st milestone



Explanations about flow charts

Flow chart (a)

Firstly, the inputs motor rate, radius, weight, height, time are obtained from the user. Then the program will give calculated outputs speed, distance, calories, steps that the user walk/ ran on the treadmill.

Flow chart (b)

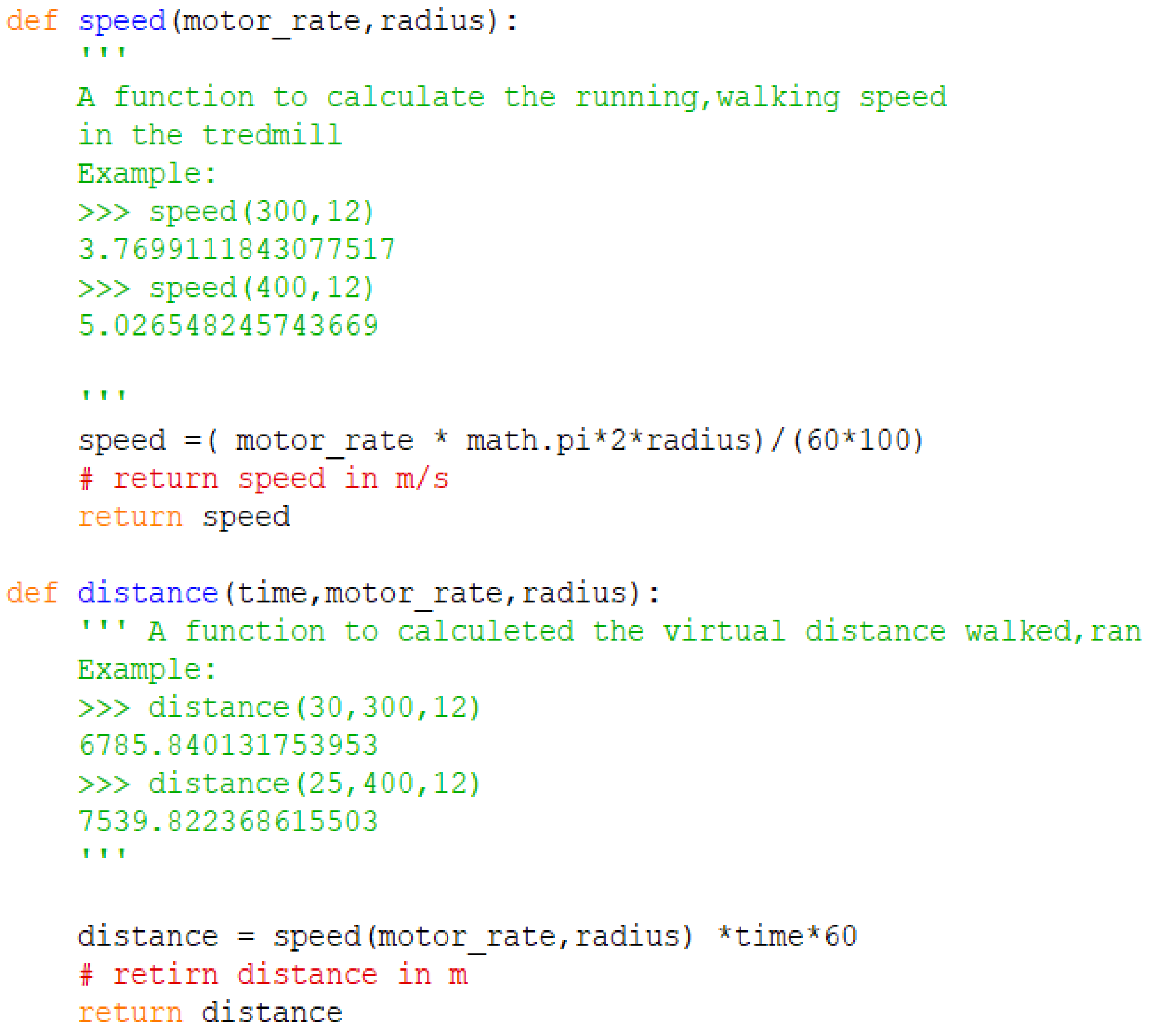
In the flow chart (b) the user needs to input motor rate, radius, weight, height then it will get the system time and it will calculate the time which the user is using the treadmill for and according to varying time it will output the speed, distance, calories, steps that the user walk/ ran on the treadmill on real time.

Note:

In the flow chart/ program the inputs motor rate, radius, weight, height can be given in various units. (value unit) as in this format. In flow chart unit conversion is only done for height. (a), (b), (c) conditions and equations in (p), (q), (r) in the block B should be changed according to the type of input.

Program code

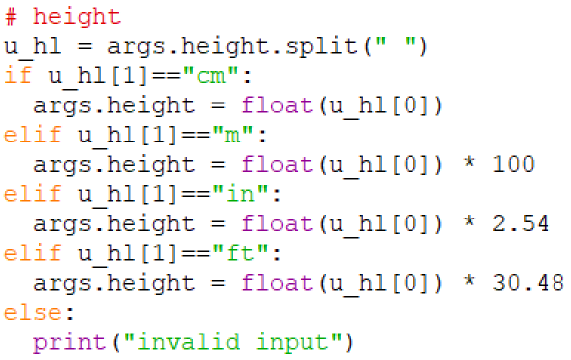
Part of the program where we defined the functions



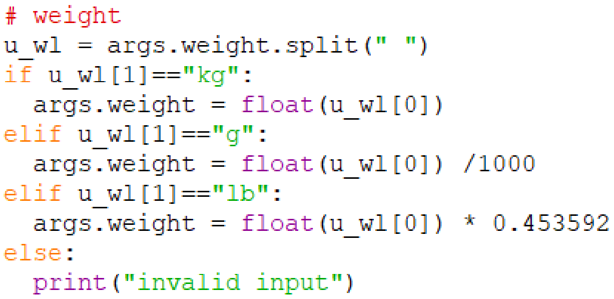
```
def speed(motor_rate,radius):
    '''
    A function to calculate the running,walking speed
    in the tredmill
    Example:
    >>> speed(300,12)
    3.7699111843077517
    >>> speed(400,12)
    5.026548245743669

    '''
    speed =( motor_rate * math.pi*2*radius)/(60*100)
    # return speed in m/s
    return speed

def distance(time,motor_rate,radius):
    ''' A function to calculeted the virtual distance walked,ran
    Example:
    >>> distance(30,300,12)
    6785.840131753953
    >>> distance(25,400,12)
    7539.822368615503
    '''

    distance = speed(motor_rate,radius) *time*60
    # retirn distance in m
    return distance
```

```
def calories(weight,height,motor_rate,radius,time):
    '''
    A function to calculate the calories burnt from begening
    Example:
    >>> calories(64,154,300,12,30)
    467.89376843225307
    >>> calories(76,162,300,12,25)
    463.01987501108374
    '''
    if speed(motor_rate,radius) <= 1.65405:
      oxygen = (0.1*speed(motor_rate,radius)*60)+ 3.5     # oxygen per minute when walking
    else:
       oxygen = (0.2*speed(motor_rate,radius)*60)+ 3.5    # oxygen per minute when running
    # calories per minute
    cal = (oxygen * weight)/200

    calories = cal * time

    return calories

def steps(time,motor_rate,radius,height):
    '''
    A function to calculate the no.of steps ran,walked
    Example:
    >>> steps(25,300,12,164)
    16697.769965338768
    '''

    do = distance(time,motor_rate,radius)

    # calculating average stride lenth in m per stride
    stride = ((height/2.54)*0.413* 2.54)/100

    # calculating no of steps
    steps = do / (stride/2)
    return steps
```

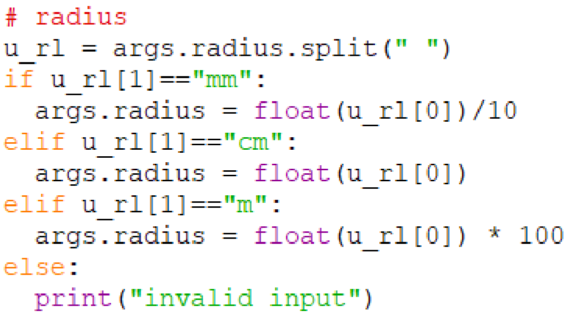
Part of the program that we used for unit conversion



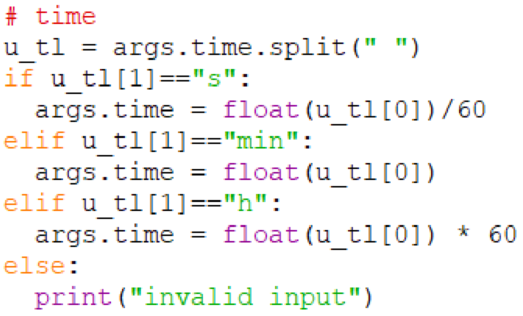
```
# height
u_hl = args.height.split(" ")
if u_hl[1]=="cm":
  args.height = float(u_hl[0])
elif u_hl[1]=="m":
  args.height = float(u_hl[0]) * 100
elif u_hl[1]=="in":
  args.height = float(u_hl[0]) * 2.54
elif u_hl[1]=="ft":
  args.height = float(u_hl[0]) * 30.48
else:
  print("invalid input")
```



```
# weight
u_wl = args.weight.split(" ")
if u_wl[1]=="kg":
  args.weight = float(u_wl[0])
elif u_wl[1]=="g":
  args.weight = float(u_wl[0]) /1000
elif u_wl[1]=="lb":
  args.weight = float(u_wl[0]) * 0.453592
else:
  print("invalid input")
```

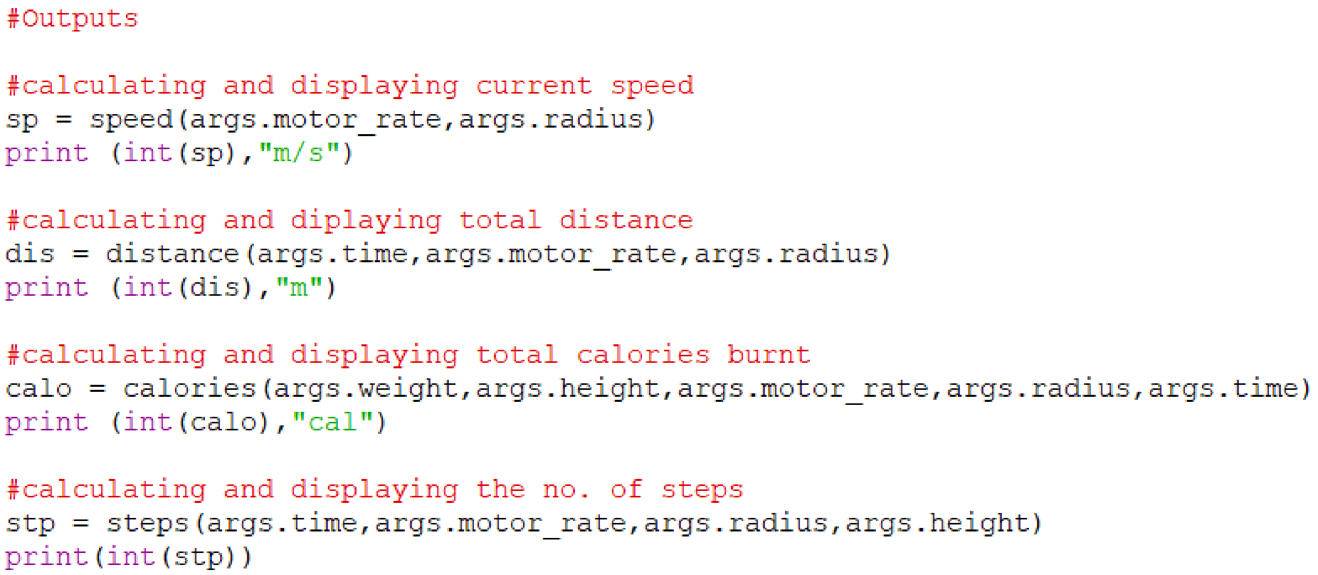


```
# radius
u_rl = args.radius.split(" ")
if u_rl[1]=="mm":
  args.radius = float(u_rl[0])/10
elif u_rl[1]=="cm":
  args.radius = float(u_rl[0])
elif u_rl[1]=="m":
  args.radius = float(u_rl[0]) * 100
else:
  print("invalid input")
```



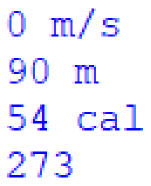
```
# time
u_tl = args.time.split(" ")
if u_tl[1]=="s":
  args.time = float(u_tl[0])/60
elif u_tl[1]=="min":
  args.time = float(u_tl[0])
elif u_tl[1]=="h":
  args.time = float(u_tl[0]) * 60
else:
  print("invalid input")
```

Part of the program that we used to get the outputs



```
#Outputs

#calculating and displaying current speed
sp = speed(args.motor_rate,args.radius)
print (int(sp),"m/s")

#calculating and diplaying total distance
dis = distance(args.time,args.motor_rate,args.radius)
print (int(dis),"m")

#calculating and displaying total calories burnt
calo = calories(args.weight,args.height,args.motor_rate,args.radius,args.time)
print (int(calo),"cal")

#calculating and displaying the no. of steps
stp = steps(args.time,args.motor_rate,args.radius,args.height)
print(int(stp))
```

Outputs we got from running the program using default parameters which were provided



```
0 m/s
90 m
54 cal
273
```

Note:

We have changed our program a little bit to run it with the default parameters which were given to us.

Equations for calculations

The equations and relevant units used in calculation are listed below.

- distance = total distance walked or ran on treadmill in meters on given time
- speed = the current, average speed of treadmill in m/s
- height = height of the user in cm
- weight = weight of the user in kg
- time = time duration that walked/ ran in minutes
- radius = radius of the motor shaft in cm
- rpm = the speed of the motor in RPM
- calories = the total calories burnt by using treadmill on time
- steps = the steps walked/ ran in treadmill

01) speed

speed = (radius/100) * (rpm/60) * 2 * pi

- speed value will give in m/s

02) distance

distance = speed * time

- distance value will give in meters.

03) calories

The equations that we have used in finding calories.

01) when walking, speed =< 3.7 mph, 1.65405 m/s

Ox = (oxygen usage mL/kg/min)

Ox = (0.1*speed*60) + (1.8 * speed * grade) + 3.5

02) when running, speed > 3.7mph, 1.65405 m/s

Ox = (oxygen usage mL/kg/min)

Ox = (0.1*speed*60) + (1.8 * speed * grade) + 3.5

NOTE:

Basically, the grade in treadmill is a measure of height distance that is for every 100 horizontal distance that walked/ran

When the treadmill is in flat/level position the grade of the treadmill will become 0.

Equation when

walking: oxygen = (0.1*speed*60) + 3.5

running: oxygen = (0.2*speed*60) + 3.5

calories per minute = (oxygen * weight) / 200

calories = calories per minute * time

= (oxygen * weight) / 200

Walking : calories = ((0.1*speed*60) + 3.5 * weight) / 200

Running : calories = ((0.2*speed*60) + 3.5 * weight) / 200

04) steps

stride = stride length of user according to height

stride = ((height/2.54) * 0.413* 2.54)/100

steps = distance / (stride/2)
= distance *100 / (((height/2.54) * 0.413* 2.54)/2)

The softwares that we used in this project.

- Python - To write the program
- Co -lab - To write the program
- Draw.io - To draw flow charts
- Word – To write the report

Assumptions

- The treadmill is used on a level/flat walking/ running purpose.
- The treadmill is working in a constant speed that was given
- When staring tread mill it starts in constant speed

Note:

The stride length like parameters are changing according to gender also. So, steps can be slightly changed. Calorie count also has various parameters rather than given inputs so it also can be slightly changed.

References

- https://towardsdatascience.com/building-a-real-time-dashboard-using-python-plotly-library-and-web-service-145f50d204f0

Calories :

- https://www.livestrong.com/article/34973-calculate-treadmill-calories/

- https://www.topendsports.com/fitness/treadmill-gradient.htm#:~:text=The%20treadmill%20grade%20is%20basically,meters%20is%20a%2015%20%25%20grade

Steps :

- https://lowellrunning.com/stepspermile/