Department of Artificial Intelligence and Data Science

Design and Analysis of Algorithms

21AD44

Assignment 1

Nitte Meenakshi Institute of Technology

Bangalore - 560064

Team Name: Uchiha

Team Members: 1) Sai Kishan V (1NT21AD056)
2) Pruthvi Krishna T S (1NT21AD037)
3) Vignesh S (1NT21AD059)
4) Rakshith R (1NT21AD041)

Selected Programming Language: Python
Team No: 7
Given Problem No: 5

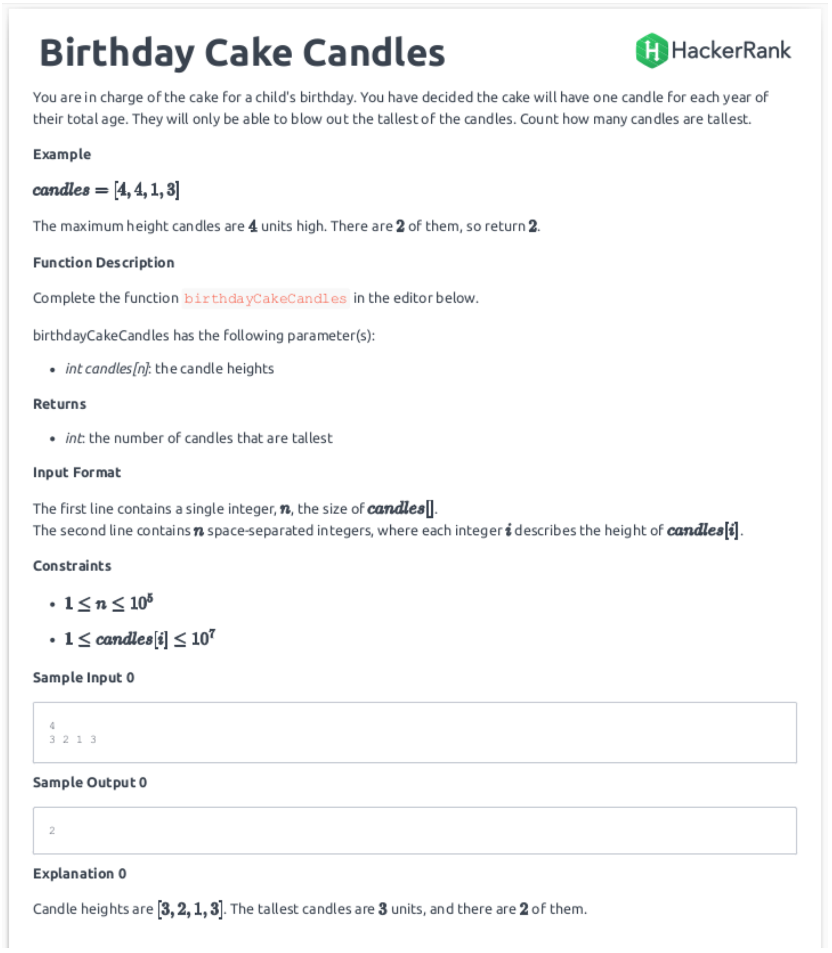
Contents

1 QUESTION:

....

You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count how many candles are tallest.

Given Instructions:



Birthday Cake Candles

HackerRank

You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count how many candles are tallest.

Example

$candles = [4, 4, 1, 3]$

The maximum height candles are 4 units high. There are 2 of them, so return 2.

Function Description

Complete the function `birthdayCakeCandles` in the editor below.

birthdayCakeCandles has the following parameter(s):

- *int candles[n]*: the candle heights

Returns

- *int*: the number of candles that are tallest

Input Format

The first line contains a single integer, n, the size of $candles[]$.
The second line contains n space-separated integers, where each integer i describes the height of $candles[i]$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq candles[i] \leq 10^7$

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

Candle heights are $[3, 2, 1, 3]$. The tallest candles are 3 units, and there are 2 of them.

2 Algorithm for the give problem:

STEP 1: Start the program.

STEP 2: Define the function birthdaycakecandles(age) that takes an age as parameter.

STEP 3: Prompt the user to enter the age and store it in the variable age.

STEP 4: Call the function birthdaycakecandles(age) with the variable age as the argument.

STEP 5: Inside the function:
- 5a: Create an empty variable(say candles) to store all candles.
- 5b: Generate random integers between 0 and age and store it in variable candles. ('Note: The random integers should be less than or equal to age')
- 5c: Iterate through the candles list and find the Tallest candle(maximum valued number)
- 5d: Return the count of the Tallest candle in candles.

STEP 6: Print the returned value

STEP 7: End the program.

3 Pseudocode:

```
function birthdaycakecandles(age):
    candles = empty list
    for i in range(0, age):
        candle = random integer between 0 and age (inclusive)
        append candle to candles
    tallestcandle = 0
    counttallestcandle = 0
    for candle in candles:
        if candle > tallestcandle:
            tallestcandle = candle
            counttallestcandle = 1
        else if candle == tallestcandle:
            counttallestcandle += 1
        else
            continue
    return candles, tallestcandle, counttallestcandle
```

4 Python code for the above Algorithm:

```
import random       # using random numbers by importing function random.

def birthdaycakecandles(age):       # defining birthdaycakecandles as a func-
tion.
        candles=[random.randint(0,age) for i in range(0,age)]       # generating
random numbers for the given age no of candles.
        return candles,max(candles),candles.count(max(candles))       # return-
ing candles, tallest candle, count of tallest candles.

while True:       #Enters the Loop without needing any condition. We used
this because of try and except.
        try:       #This executes when the right input is given.
              age=int(input("Enter the Age:"))       # prompt ur age.
              candles,tallest_candle,count_tallest_candle=birthdaycakecandles(age)
      # calling birthdaycandles and storing returned values from birthdaycake-
candles.

              print("Candles:",candles)       # printing all the candles.
              print("Tallest candle:",tallest_candle)       # printing the tallest can-
dle.
              print("You have to blow:",count_tallest_candle,"candles")       # print-
ing the count of the tallest candle.
              exit()       # Exiting from the loop.

        except ValueError:       #This executes when the wrong input is given.
              print("Please enter the positive integer only!!")
```

4.1 Working of the above Python code:

1. Firstly we ask the user to input the age and store it in a variable called 'age'.
2. Now this birthdaycakecandles function is called by sending 'age' as a argument.
3. Now it goes to the birthdaycakecandles function by taking 'age' as parameter.

    3a. Here we are generating random numbers for the given 'age' number of candles. And making sure that all the random numbers generated will be less than or equal to the 'age'. And storing it in a variable called 'candles'.
    Here we are using 'List comprehension method' so that the time taken will be reduced rather than normal 'append method' where we need to append each and every time when a random number is generated which takes more time. Therefore, we used 'List comprehension'.

    3b. Now we are returning three things:
    a)The 'candles list' which contains all the random number generated.
    b)The Maximum number present in the list.
    c)The count of the Maximum number.

4. The returned values are stored in three variables. (a)'candles' to store all the random numbers generated, (b)'tallest_candle' to store the maximum value, (c)'count_tallest_candle' to store the count of the tallest candle.

5. Finally we are printing all the values using 'print()' statement.

Now we used 'try and except' so that if any input given other than 'positive integers' it tells us to correct the input.
Since we 'infinite while loop' it keeps asking for the input until you give the correct input.

4.2 Data-structure used: List

1. Dynamic Size: Lists in Python are dynamically sized, meaning they can grow or shrink as needed. This allows you to add or remove candles from the list based on the age input. If you need to accommodate different ages and varying numbers of candles, a list is well-suited for this purpose.

2. Compatibility with List Comprehension: If you decide to enhance the code by using list comprehension to generate the candle heights or perform other operations, having a list data structure already in place would align well with such enhancements. List comprehensions are a concise and efficient way to generate or manipulate lists based on some logic.

3. By using List Comprehension we can reduce the time of execution i.e., generating random numbers would be faster than normal method.

5 Asymptotic Time Complexity Analysis:

The devised Python code snippet does not have explicit loops or conditional statements that would result in different best case, worst case, or average case time complexities. Therefore, we can assume that the time complexity remains the same regardless of the input values.

The time complexity of the code is O(age) in all cases. This means that the running time of the code grows linearly with the size of the input, which is represented by the variable ”age” in the code.

In the Best case scenario: where ”age” is small, the code will still have a linear time complexity of O(age), but the actual running time will be relatively faster compared to larger values of ”age”.

In the Worst case scenario: where ”age” is very large, the code will still have a linear time complexity of O(age), but the running time will be proportionally longer compared to smaller values of ”age”.

Similarly, for the Average case: the time complexity remains O(age), indicating that the running time scales linearly with the input size ”age” on average.

Therefore, to summarize, the time complexity of the given code is O(age) in all cases: best case, worst case, and average case, Where ”age” is the input given.

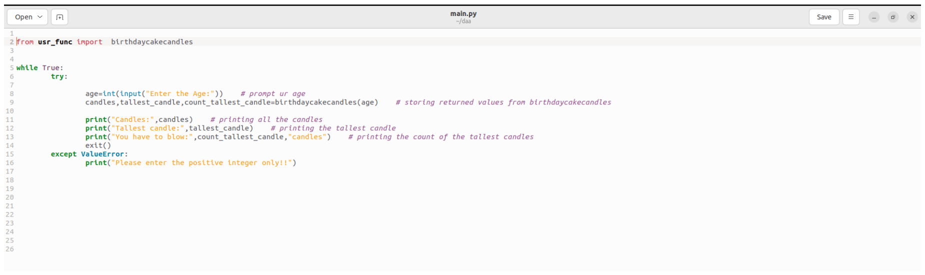
6 Makefile:

In Python, a Makefile is a file used to define a set of commands and dependencies to automate the build process or perform other tasks. While Makefiles are traditionally associated with C/C++ projects, they can also be used in Python projects to manage tasks such as running tests, generating documentation, cleaning up files, and more.

We used 'Makefile' to build and run this project. To automate the run process because we used two separate files 'one for main function' and 'another for user defined function i.e., birthdaycakecandle()'. Therefore, Makefile made this process easier which implicitly runs all the files.

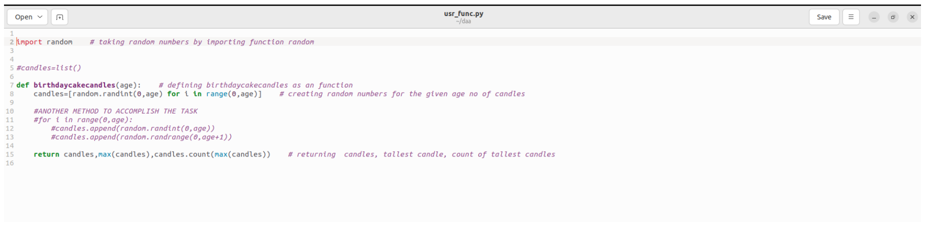
Here are steps:
STEP 1: First we created three separate files, one for main function named as 'main.py' another is our own function birthdaycakecandles() named as 'usr_func.py'.

The main.py contains:



```python

from usr_func import  birthdaycakecandles


while True:
        try:

                age=int(input("Enter the Age:"))    # prompt ur age
                candles,tallest_candle,count_tallest_candle=birthdaycakecandles(age)    # storing returned values from birthdaycakecandles

                print("Candles:",candles)    # printing all the candles
                print("Tallest candle:",tallest_candle)    # printing the tallest candle
                print("You have to blow:",count_tallest_candle,"candles")    # printing the count of the tallest candles
                exit()
        except ValueError:
                print("Please enter the positive integer only!!")
```

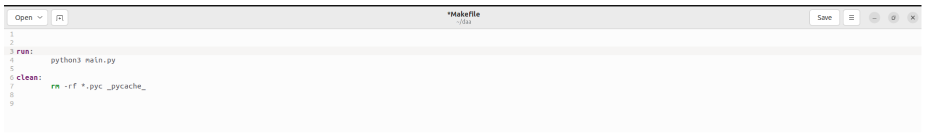
The usr_func.py contains:



```python

import random    # taking random numbers by importing function random


#candles=list()

def birthdaycakecandles(age):    # defining birthdaycakecandles as an function
    candles=[random.randint(0,age) for i in range(0,age)]    # creating random numbers for the given age no of candles

    #ANOTHER METHOD TO ACCOMPLISH THE TASK
    #for i in range(0,age):
        #candles.append(random.randint(0,age))
        #candles.append(random.randrange(0,age+1))

    return candles,max(candles),candles.count(max(candles))    # returning  candles, tallest candle, count of tallest candles

```

STEP 2: Next with the help of 'text editor' we created another file named as 'Makefile'(without any file extension). Which contains two typical commands: (i) The 'run' command to run the python code, (ii) the clean command to clean pycache.

The Makefile contains:



```


run:
	python3 main.py

clean:
	rm -rf *.pyc _pycache_


```

run command:
-This is used to run the specified target.
-Here we specified 'main.py' so it runs the main.py code, which intern runs the usr_func.py code.
-So finally, we can run all these codes just be typing 'make run'

clean command:
-This command is used to remove any generated or temporary files.
-This command removes all .pyc files (compiled Python files) and the __pycache__directory recursively from the current directory.

STEP 3: So, after creating all these three files we have to put all these three files in one single directory (say daa).

So, finally our code is ready to be compiled using Makefile.

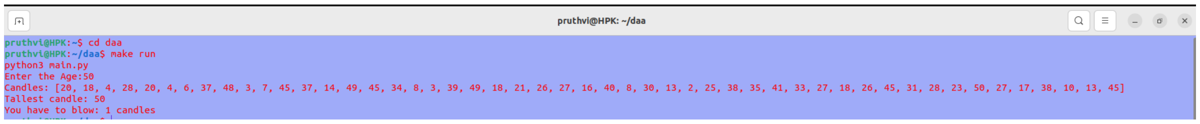
How to get output:

STEP 1: Go to terminal and navigate to the directory(daa) where all these three files are present.
STEP 2: Type 'make run'.

If you want to remove all the unwanted files generated by python, just type 'make clean'.

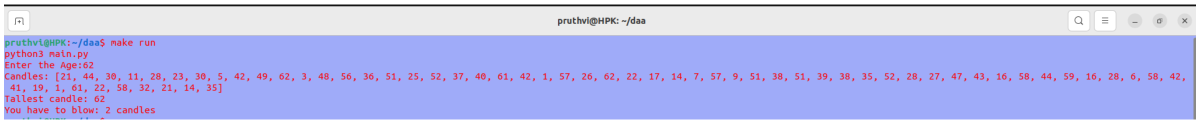
7 Test cases:

Test case1 1:



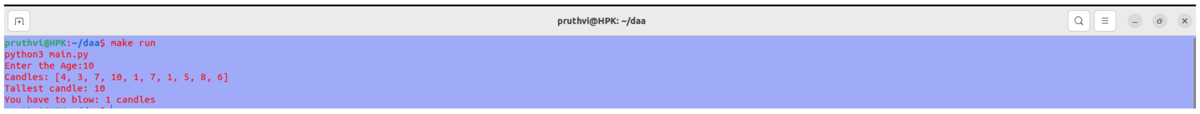
```
pruthvi@HPK: ~/daa
pruthvi@HPK:~$ cd daa
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:50
Candles: [20, 18, 4, 28, 20, 4, 6, 37, 48, 3, 7, 45, 37, 14, 49, 45, 34, 8, 3, 39, 49, 18, 21, 26, 27, 16, 40, 8, 30, 13, 2, 25, 38, 35, 41, 33, 27, 18, 26, 45, 31, 28, 23, 50, 27, 17, 38, 10, 13, 45]
Tallest candle: 50
You have to blow: 1 candles
```

Test case 2:



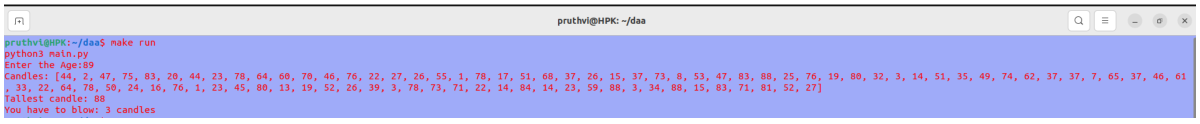
```
pruthvi@HPK: ~/daa
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:62
Candles: [21, 44, 30, 11, 28, 23, 30, 5, 42, 49, 62, 3, 48, 56, 36, 51, 25, 52, 37, 40, 61, 42, 1, 57, 26, 62, 22, 17, 14, 7, 57, 9, 51, 38, 51, 39, 38, 35, 52, 28, 27, 47, 43, 16, 58, 44, 59, 16, 28, 6, 58, 42,
 41, 19, 1, 61, 22, 58, 32, 21, 14, 35]
Tallest candle: 62
You have to blow: 2 candles
```

Test case 3:



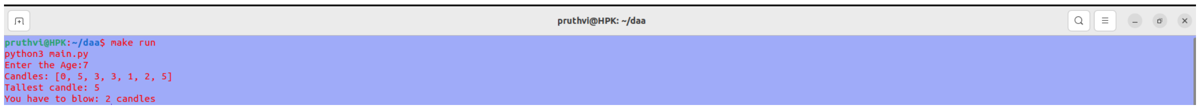
```
pruthvi@HPK: ~/daa
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:10
Candles: [4, 3, 7, 10, 1, 7, 1, 5, 8, 6]
Tallest candle: 10
You have to blow: 1 candles
```

Test case 4:



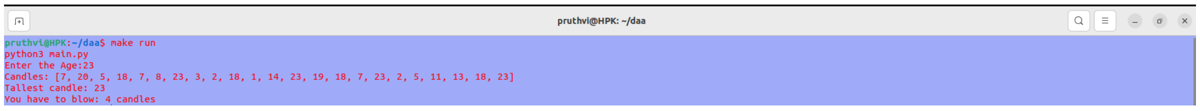
```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:89
Candles: [44, 2, 47, 75, 83, 20, 44, 23, 78, 64, 60, 70, 46, 76, 22, 27, 26, 55, 1, 78, 17, 51, 68, 37, 26, 15, 37, 73, 8, 53, 47, 83, 88, 25, 76, 19, 80, 32, 3, 14, 51, 35, 49, 74, 62, 37, 37, 7, 65, 37, 46, 61
, 33, 22, 64, 78, 50, 24, 16, 76, 1, 23, 45, 80, 13, 19, 52, 26, 39, 3, 78, 73, 71, 22, 14, 84, 14, 23, 59, 88, 3, 34, 88, 15, 83, 71, 81, 52, 27]
Tallest candle: 88
You have to blow: 3 candles
```

Test case 5:



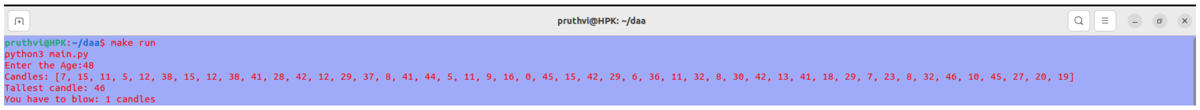
```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:7
Candles: [0, 5, 3, 3, 1, 2, 5]
Tallest candle: 5
You have to blow: 2 candles
```

Test case 6:



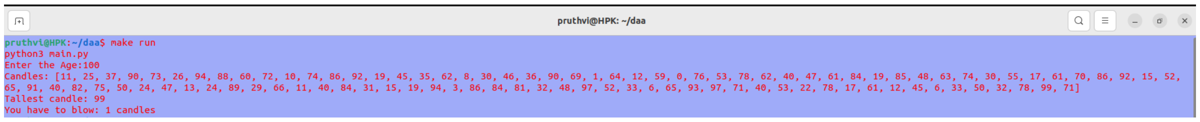
```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:23
Candles: [7, 20, 5, 18, 7, 8, 23, 3, 2, 18, 1, 14, 23, 19, 18, 7, 23, 2, 5, 11, 13, 18, 23]
Tallest candle: 23
You have to blow: 4 candles
```

Test case 7:



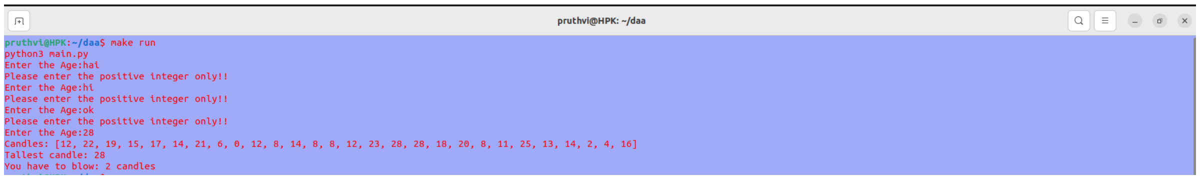
```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:48
Candles: [7, 15, 11, 5, 12, 38, 15, 12, 38, 41, 28, 42, 12, 29, 37, 8, 41, 44, 5, 11, 9, 16, 0, 45, 15, 42, 29, 6, 36, 11, 32, 8, 30, 42, 13, 41, 18, 29, 7, 23, 8, 32, 46, 10, 45, 27, 20, 19]
Tallest candle: 46
You have to blow: 1 candles
```

Test case 8:



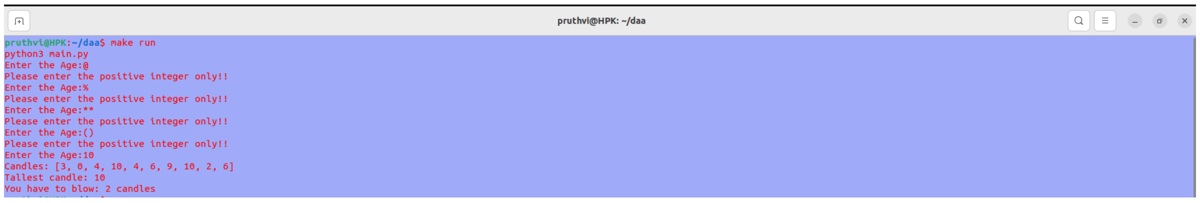
```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:100
Candles: [11, 25, 37, 90, 73, 26, 94, 88, 60, 72, 10, 74, 86, 92, 19, 45, 35, 62, 8, 30, 46, 36, 90, 69, 1, 64, 12, 59, 0, 76, 53, 78, 62, 40, 47, 61, 84, 19, 85, 48, 63, 74, 30, 55, 17, 61, 70, 86, 92, 15, 52,
65, 91, 40, 82, 75, 50, 24, 47, 13, 24, 89, 29, 66, 11, 40, 84, 31, 15, 19, 94, 3, 86, 84, 81, 32, 48, 97, 52, 33, 6, 65, 93, 97, 71, 40, 53, 22, 78, 17, 61, 12, 45, 6, 33, 50, 32, 78, 99, 71]
Tallest candle: 99
You have to blow: 1 candles
```

Test case 9:



```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:hai
Please enter the positive integer only!!
Enter the Age:hi
Please enter the positive integer only!!
Enter the Age:ok
Please enter the positive integer only!!
Enter the Age:28
Candles: [12, 22, 19, 15, 17, 14, 21, 6, 0, 12, 8, 14, 8, 8, 12, 23, 28, 28, 18, 20, 8, 11, 25, 13, 14, 2, 4, 16]
Tallest candle: 28
You have to blow: 2 candles
```

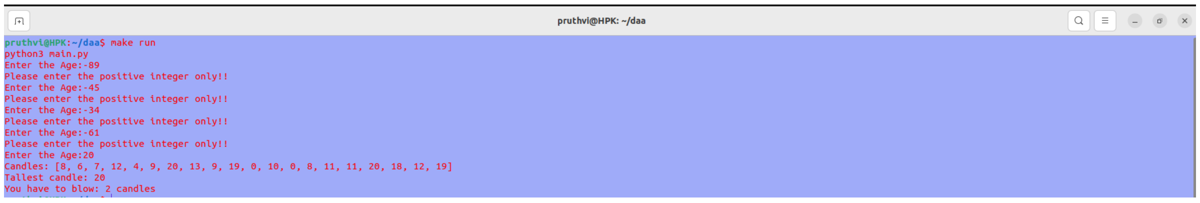
Test case 10:



pruthvi@HPK: ~/daa

```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:@
Please enter the positive integer only!!
Enter the Age:%
Please enter the positive integer only!!
Enter the Age:**
Please enter the positive integer only!!
Enter the Age:()
Please enter the positive integer only!!
Enter the Age:10
Candles: [3, 0, 4, 10, 4, 6, 9, 10, 2, 6]
Tallest candle: 10
You have to blow: 2 candles
```

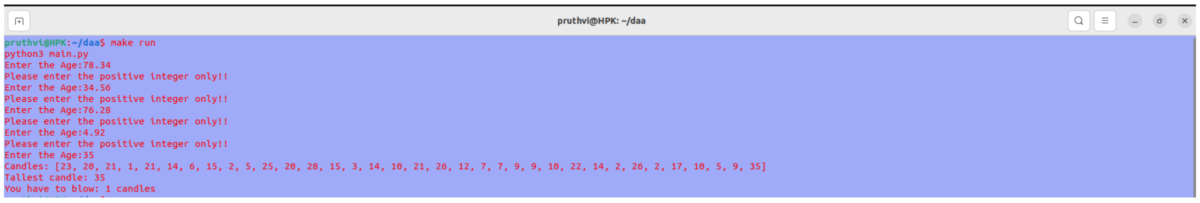
Test case 11:



pruthvi@HPK: ~/daa

```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:-89
Please enter the positive integer only!!
Enter the Age:-45
Please enter the positive integer only!!
Enter the Age:-34
Please enter the positive integer only!!
Enter the Age:-61
Please enter the positive integer only!!
Enter the Age:20
Candles: [8, 6, 7, 12, 4, 9, 20, 13, 9, 19, 0, 10, 0, 8, 11, 11, 20, 18, 12, 19]
Tallest candle: 20
You have to blow: 2 candles
```

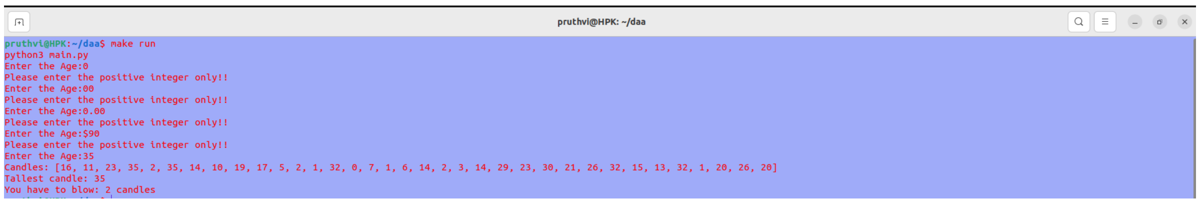
Test case 12:



pruthvi@HPK: ~/daa

```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:78.34
Please enter the positive integer only!!
Enter the Age:34.56
Please enter the positive integer only!!
Enter the Age:76.28
Please enter the positive integer only!!
Enter the Age:4.92
Please enter the positive integer only!!
Enter the Age:35
Candles: [23, 20, 21, 1, 21, 14, 6, 15, 2, 5, 25, 20, 28, 15, 3, 14, 10, 21, 26, 12, 7, 7, 9, 9, 10, 22, 14, 2, 26, 2, 17, 10, 5, 9, 35]
Tallest candle: 35
You have to blow: 1 candles
```

Test case 13:



pruthvi@HPK: ~/daa

```
pruthvi@HPK:~/daa$ make run
python3 main.py
Enter the Age:0
Please enter the positive integer only!!
Enter the Age:00
Please enter the positive integer only!!
Enter the Age:0.00
Please enter the positive integer only!!
Enter the Age:$90
Please enter the positive integer only!!
Enter the Age:35
Candles: [16, 11, 23, 35, 2, 35, 14, 10, 19, 17, 5, 2, 1, 32, 0, 7, 1, 6, 14, 2, 3, 14, 29, 23, 30, 21, 26, 32, 15, 13, 32, 1, 20, 26, 20]
Tallest candle: 35
You have to blow: 2 candles
```

Thank you!!