COMPUTER SCIENCE

PROJECT FILE

MADE BY :- Gauri Gupta

CLASS :- XI A (2020-2021)

ROLL NO. :- 13

STUDENT MANAGEMENT SYSTEM

ACKNOWLEDGEMENT

We are greatly indebted to our teacher,

Mrs. Urvashi Singhal

who throughout the tedious process of completion of the project displayed with full patience and helped me tide over the obstacles faced during this course by providing me with the necessary council and encouragement at each step and pointed out my various shortcomings in tackling this project.

Student’s Signature
Name:…………….

Mrs. Urvashi Singhal
(HOD Computer Science)

CERTIFICATE

This is to certify that

GAURI GUPTA

and

RHEA , HANSITA

of
Class XI A, Ahlcon International School,
have worked on the project titled

STUDENT MANAGEMENT

and
have completed the project to my
satisfaction.

Mrs. Urvashi Singhal
HOD Computer Science

Overview Of Python...

- Python was developed in the late eighties, i.e., the late 1980's by **Guido van Rossum** at the **National Research Institute for Mathematics and Computer Science** in the Netherlands as a successor of ABC language capable of exception handling and interfacing.
- Python is derived from programming languages such as ABC, Modula 3, small talk, Algol-68.
- Van Rossum picked the name Python for the new language from a TV show, Monty Python's Flying Circus.
- Python page is a file with a **.py** extension that contains could be the combination of HTML Tags and Python scripts.
- In December 1989, the creator developed the 1st python interpreter as a hobby, and then on 16 October 2000, Python 2.0 was released with many new features.
- On 3rd December 2008, **Python 3.0** was released with more testing and included new features.
- Python is an open-source scripting language.
- Python is open-source, which means that anyone can download it freely from **www.python.org** and use it to develop programs. Its source code can be accessed and modified as required in the project.
- Python is one of the official languages at Google

Aim and need for this program

1) Readable and Maintainable Cod

While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

2) Multiple Programming Paradigms

Like other modern programming languages, Python also supports several programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

3) Compatible with Major Platforms and Systems

At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

4) Robust Standard Library

Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

5) Many Open Source Frameworks and Tools

As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you can accelerate desktop GUI application development using **Python GUI frameworks** and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython.

6) Simplify Complex Software Development

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many **Python developers** even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

7) Adopt Test Driven Development

You can use Python to create prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

However, Python, like other programming languages, has its own shortcomings. It lacks some of the built-in features provided by other modern programming language. Hence, you have to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is slower than several widely used programming languages including Java and C++. You have to speed up the Python application by making changes to the application code or using custom runtime. But you can always use Python to speed up software development and simplify software maintenance.

Hardware And Software Requirements…

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer.

The **hardware requirements** are the **requirements** of a **hardware** device. Most **hardware** only has operating **system requirements** or compatibility. For example, a printer may be compatible with Windows XP but not compatible with newer versions of Windows like Windows 10, Linux, or the Apple macOS.

A software requirement can be of 3 types:

- Functional requirements.
- **Non**-functional requirements.
- Domain requirements.

Hardware Requirements

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.
- Sound card w/speakers.
- Some classes require a camera and microphone.

Software Requirements Characteristics

- Clear.
- Correct.
- Consistent.
- Coherent.
- Comprehensible.
- Modifiable.
- Verifiable.
- Prioritized

FUNCTIONS

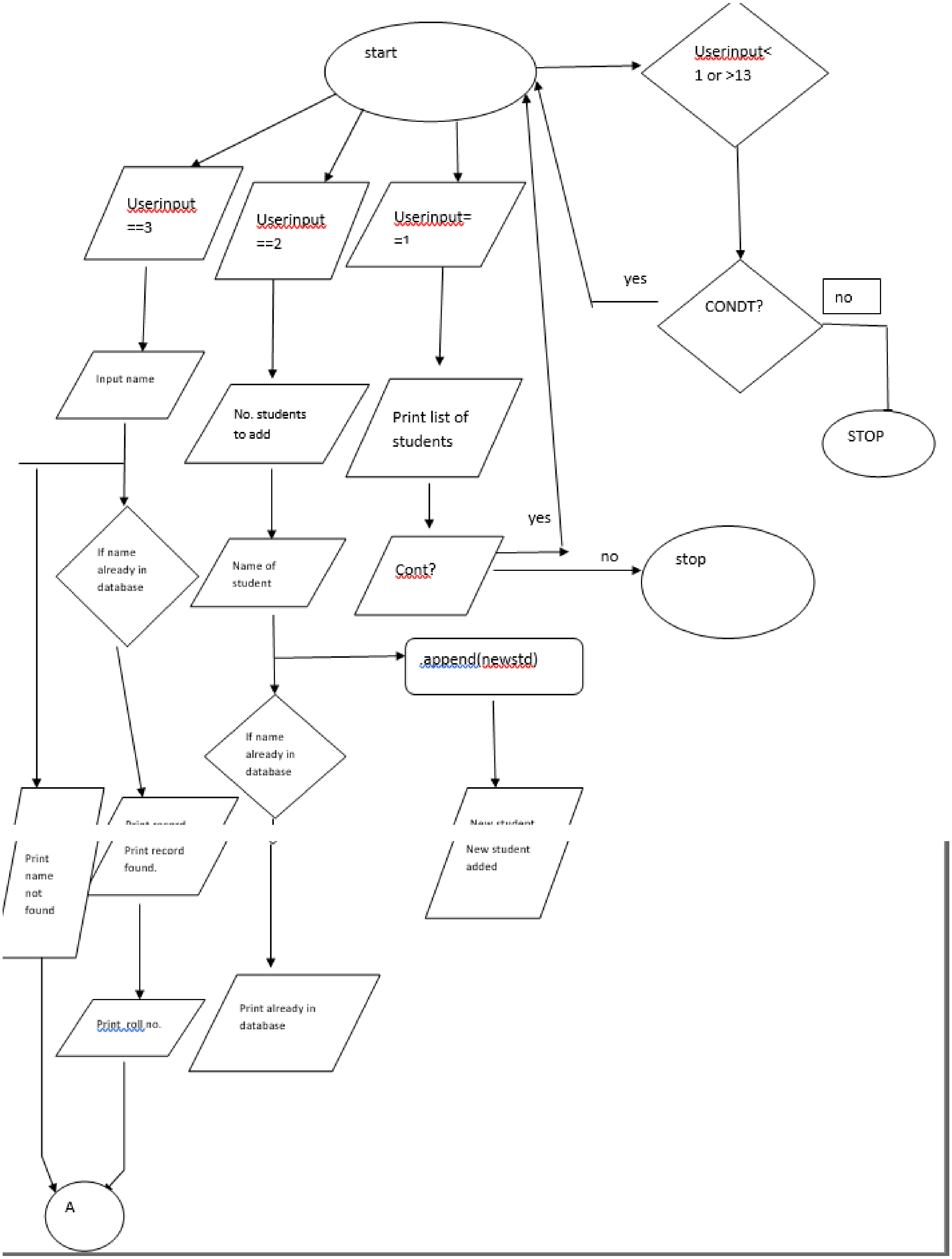
Print()
For
Int (())
Format()
Or
List()
False

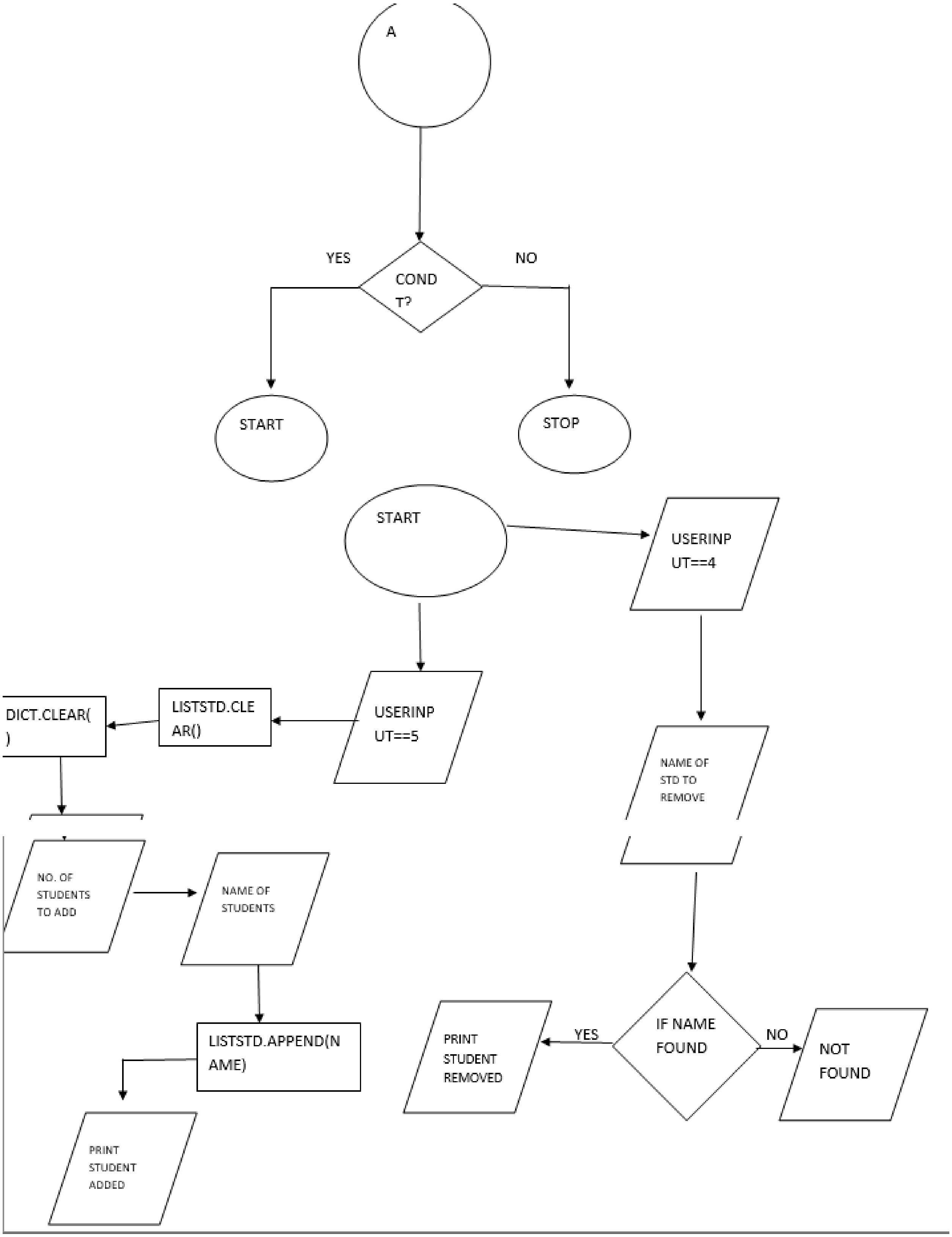
Global
in
Input()
Elif:
Exit()
Sorted()
def __()

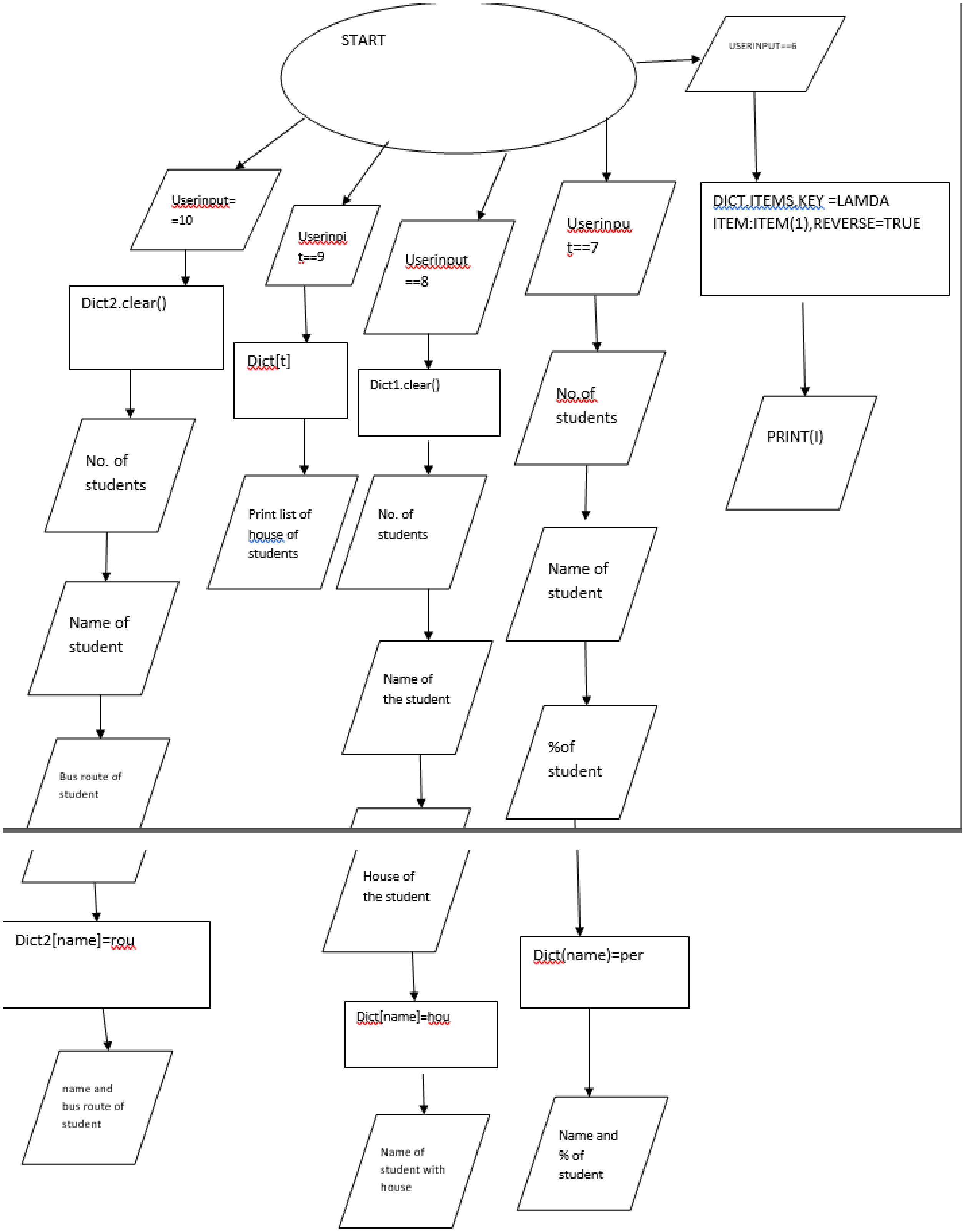
Dict[]
Import
If:
Else:
Range()
True

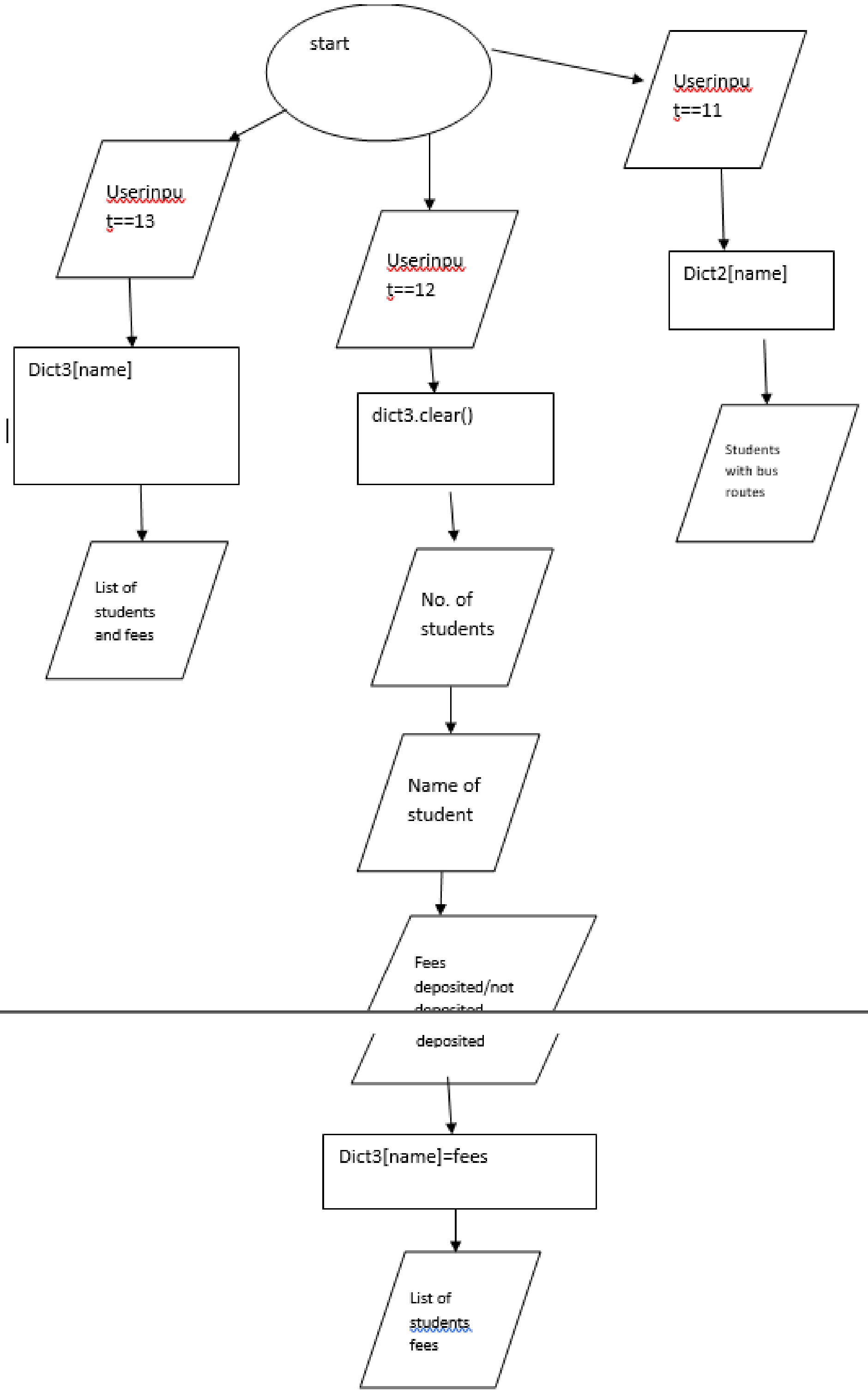
FLOW CHART OF PROGRAM

(next page)


start
Userinput< 1 or >13
Userinput ==3
Userinput ==2
Userinput= =1
yes
CONDT?
no
STOP
Input name
No. students to add
Print list of students
If name already in database
Name of student
yes
Cont?
no
stop
append(newstd)
If name already in database
Print record found.
New student added
Print name not found
Print roll no.
Print already in database
A


A
YES
COND
T?
NO
START
STOP
START
USERINP
UT==4
USERINP
UT==5
LISTSTD.CLE
AR()
DICT.CLEAR(
)
NAME OF
STD TO
REMOVE
NO. OF
STUDENTS
TO ADD
NAME OF
STUDENTS
LISTSTD.APPEND(N
AME)
PRINT
STUDENT
ADDED
PRINT
STUDENT
REMOVED
YES
IF NAME
FOUND
NO
NOT
FOUND


START
USERINPUT==6
DICT.ITEMS.KEY =LAMDA ITEM:ITEM(1),REVERSE=TRUE
PRINT(I)
Userinput= =10
Dict2.clear()
No. of students
Name of student
Bus route of student
Dict2[name]=rou
name and bus route of student
Userinpt==9
Dict[t]
Print list of house of students
Userinput ==8
Dict1.clear()
No. of students
Name of the student
House of the student
Dict[name]=hou
Name of student with house
Userinput==7
No.of students
Name of student
%of student
Dict(name)=per
Name and % of student


start
Userinput==13
Userinput==12
Userinput==11
Dict3[name]
dict3.clear()
Dict2[name]
List of students and fees
No. of students
Students with bus routes
Name of student
Fees deposited/not deposited
deposited
Dict3[name]=fees
List of students fees

MAIN SOURCE CODE

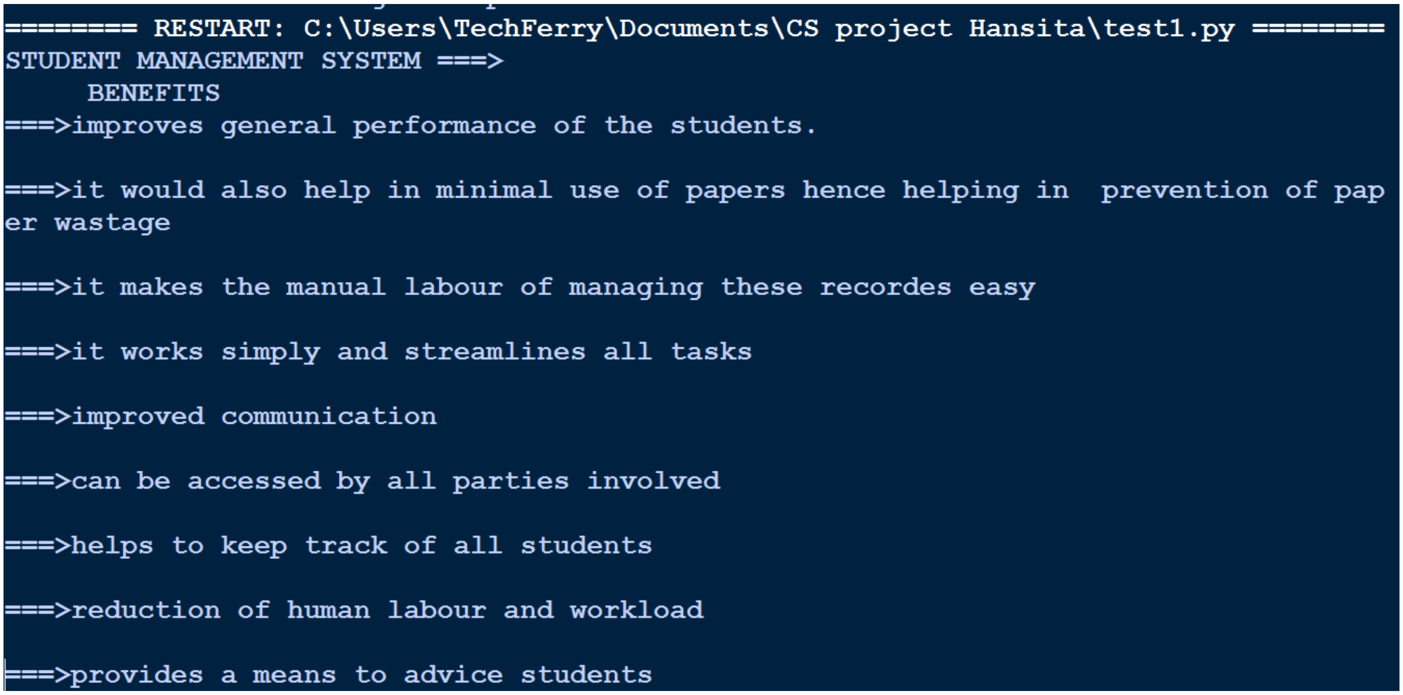
```
print('STUDENT MANAGEMENT SYSTEM ===>\

\n    BENEFITS\

\n===>improves general performance of the students.\

\n \

\n===>it would also help in minimal use of papers hence helping in  prevention of paper wastage\

\n \

\n===>it makes the manual labour of managing these recordes easy\

\n \

\n===>it works simply and streamlines all tasks\

\n \

\n===>improved communication\

\n \

\n===>can be accessed by all parties involved\

\n \

\n===>helps to keep track of all students\

\n\

\n===>reduction of human labour and workload\

\n\

\n===>provides a means to advice students\

\n\

\n===>improves the general comfort of the staff\

\n\
\n \====-HELPFUL BENFITS OF PYTHON-=========\
\n ====>it is versatile\
\n\====>easy to use\
\n\===>it is fast to develop\
\n\===> it is open source with a vibrant community\
\n\===>it has all the libraries you can imagine\
\n\===>great prototypes: you can do more by coding less')

global liststd
# creating a variavle inside a function
#making ListStd As super Global Variable i.e
# a variable inside and outide the functions

liststd= ['a','b','c', 'd','e','f','g','h',\
              'i','j','k','l','m']

global dict
# mutable data type
dict = {}
for name in liststd:
   dict[name] = 0

global dict1
dict1 = {}
for name in liststd:
    dict1[name] = ""


global dict2

dict2 = {}

for name in liststd:

    dict2[name] = ""


global dict3

dict3 = {}

for name in liststd:

    dict3[name] = ""




"""some dictionaries have been defined as

global in order to make use of it in the whole program"""


def main():


    #have to store the whole program so in 'main' so that it can run again


    import os


    #user and software interface

    import platform
```

```
# to see the underlying platforms info or data




        # list of students

def managestudent():
    # function for the student management system

    x= "#" * 30
    y= "=" * 28
    global bye
    bye='===>this program has ended<==='

print('''
----------------------------------------------------------
|----------------------------------------------------------|
===============STUDENT MANAGEMENT FILE=====================
|-----------------------------------------------------------|

______welcome_______

enter 1: to view students' list
enter 2: to add new student
enter 3: to search student

enter 4: to remove student

enter 5: to make a new list of students

enter 6: to view percentage of students

enter 7: to enter percentage of student

enter 8: to enter house of student

enter 9: to view house of student

enter 10: to enter bus routes of students

enter 11: to view bus routes of students

enter 12: to record entries of deposition of fees

enter 13: to view entries of deposition of fees

 ''')


#options for the user to choose from^


userinput=int(input('select the one of the given options above :'))

# user will select any of the options




if(userinput==1) :


    print('list students\n')


    #will print the given sentence in the new line
```

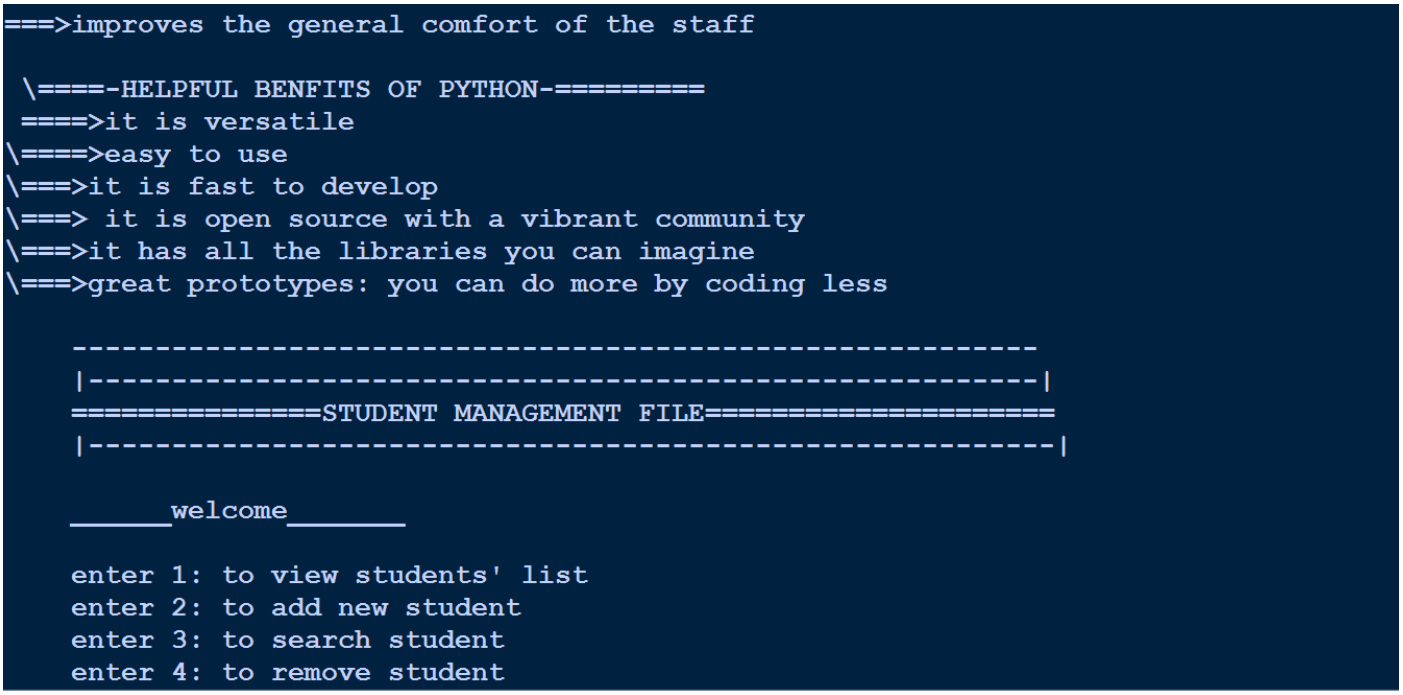
```
  for students in liststd:
    print("- {}" . format(students))
      #it will print the list of students in the curly bracket



  j = input("Do you want to continue (YES / NO) - ").lower()
  #the input would be converted to lower case
  if j == 'YES' or j == 'yes' or j == 'Yes':
    main()
  #the program would be returned to the start
  else:
    print('''===============================================
       ==================-system end-====================
       ==========-thank you for using our system-========
       ================================================''')
    exit()
   #asking the user to start again or end the program




elif(userinput==2):

  #this option adds a new line
  g = int(input("Enter number of students you want to add in the list - "))
  for i in range (0,g):
    newStd = input('enter new student : ')

    l2 = list(newStd)


    if(newStd in liststd):

        #this condition would check if you are in the list before hand

        print('\n this student {} already in the database'.format(newStd))

            #error message

    else:

        liststd.append(newStd)

        #add the student int he existing list

        print('\n=> new student {} successfully add \n'.format(newStd))

        #.format would print the names in the curly bracket


for students in liststd:

    print('=. {}'.format(students))




j = input("Do you want to continue (YES / NO) - ").lower()

if j == 'YES' or j == 'yes' or j == 'Yes':

    main()

else:

    print('''===============================================

    ==================-system end-====================

    ==========-thank you for using our system-========
        ================================================''')

        exit()

                #asking the user to end the program or continue the program




elif(userinput==3):


    #this option will search student from the list

        srcStd= input('enter student name in search:')

        if(srcStd in liststd):

            #this condition searching the student

            print('\n=> record found of student {}'. format(srcStd))

            print("Roll no. - ",liststd.index(srcStd) + 1)

        else:

            print('\n => no record found of student {}'. format(srcStd))

            # error message




        j = input("Do you want to continue (YES / NO) - ").lower()

        if j == 'YES' or j == 'yes' or j == 'Yes':

            main()

        else:

            print('''================================================

                ==================-system end-====================

                ==========-thank you for using our system-========
                    ========================================================''')
                exit()
                    # asking the user to end or continue the program





elif(userinput==4):

    #this option will remove student from the list
    rmStd= input('enter student name to remove : ')
    if (rmStd in liststd):
        # condition for removing the student from the list
        liststd.remove(rmStd)
        print('\n => student {} successfully deleted \n'.format(rmStd))
    else:
        print('student not found in the data base')



    j = input("Do you want to continue (YES / NO) - ").lower()
    if j == 'YES' or j == 'yes' or j == 'Yes':
        main()
    else:
        print('''=====================================================
                 ==================-system end-====================
                  ==========-thank you for using our system-========

                  ================================================''')

        exit()

            #asking the user to continue or end the program




elif(userinput == 5):


    liststd.clear()

    #.clear function would remove all elements in the list

    dict.clear()

    #would clear the dictionary


    k = int(input("Enter number of students to be added in the list - "))

    for i in range (0,k):

        name = input("Enter name of student - ")

        liststd.append(name)

        #append function would add students in list

        print("Added\n")

    print("\nNew list = ",liststd)

    """ a new list has been created"""

    for name in liststd:

        dict[name] = 0

        #name would be in place of index 0
        j = input("Do you want to continue (YES / NO) - ").lower()

        # .lower : any name that the userinputs would be in lowercase

        if j == 'YES' or j == 'yes' or j == 'Yes':

            main() #main() is defined in the beginning of the code and would restart the code from the
begining

        else:

            print('''================================================

                  ==================-system end-====================

                 ==========-thank you for using our system-========

                 =================================================''')

            exit()

            #the code would end with exit(), it wsould exit the code

         #asking the user to continue or end the program




    elif(userinput==6):



        for i in sorted(dict.items(), key=lambda item: item[1],reverse = True):

             print(i)

             #sorted in descending order

            #lamda is a small function which can take any arguement or form but of only one expression

        j = input("Do you want to continue (YES / NO) - ").lower()
    if j == 'YES' or j == 'yes' or j == 'Yes':

        main()

    else:

            print('''===============================================

                      ==================-system end-====================

                    ==========-thank you for using our system-========

                    ================================================''')

            exit()





elif(userinput==7):



# create an empty list

# Add student information to the list



    count = int(input("Enter no. of students - "))

    for i in range(0,count):

        name = input("Enter student name -  ")

        per = int(input("Enter % -  "))

        dict[name] = per

      #dictionary has been created
    j = input("Do you want to continue (YES / NO) - ").lower()

    if j == 'YES' or j == 'yes' or j == 'Yes':

        main()

    else:

        print('''===============================================

              ==================-system end-====================

              ==========-thank you for using our system-========

             ================================================''')

        exit()

            #asking the user to continue or end the program




elif(userinput==8):


    dict1.clear()

    #dictionary has been cleared

    ct = int(input("Enter no. of students - "))

    for p in range(0,ct):

        name = input("\nEnter student name -  ")

        hou = input("Enter name of house -  ")

        dict1[name] = hou

        #this will print name of the student and house of the student as dictioanry has been created
    j = input("Do you want to continue (YES / NO) - ").lower()

    if j == 'YES' or j == 'yes' or j == 'Yes':

        main()

    else:

        print('===============================================\

              ==================-system end-====================\

              ==========-thank you for using our system-========\

              ================================================')

        exit()




elif(userinput==9):


    for t in dict1:

        print(t, '-',dict1[t])

    #the names and houses are stored in 'dict1' so it will print the list of students and their respective houses

    j = input("Do you want to continue (YES / NO) - ").lower()

    if j == 'YES' or j == 'yes' or j == 'Yes':

        main()

    else:

        print('===============================================\

              ==================-system end-====================\

              ==========-thank you for using our system-========\
              ================================================')
        exit()


elif(userinput==10):
    dict2.clear()
    ct = int(input("Enter no. of students - "))
    for p in range(0,ct):
        name = input("\nEnter student name -  ")
        rou = input("Enter bus route -  ")
        dict2[name] = rou
        #name and rou have been stored in dict2 and would print together


    j = input("Do you want to continue (YES / NO) - ").lower()
    if j == 'YES' or j == 'yes' or j == 'Yes':
        main()
    else:
        print('===============================================\
              ==================-system end-====================\
              ==========-thank you for using our system-========\
              ================================================')
        exit()



elif(userinput==11):
    for name in dict2:
        print(name, '-',dict2[name])

    j = input("Do you want to continue (YES / NO) - ").lower()
    if j == 'YES' or j == 'yes' or j == 'Yes':
        main()
    else:
        print('==============================================\
            ==================-system end-====================\
            ==========-thank you for using our system-========\
            ===============================================')
    exit()



elif(userinput==12):

    dict3.clear()
    ct = int(input("Enter no. of students - "))
    for p in range(0,ct):
        name = input("\nEnter student name -  ")
        fees = input("Fees (deposited/not deposited): -  ")
        dict3[name] = fees

    j = input("Do you want to continue (YES / NO) - ").lower()
```

```
        if j == 'YES' or j == 'yes' or j == 'Yes':
            main()
        else:
            print('===============================================\
                ==================-system end-====================\
                ==========-thank you for using our system-========\
                ================================================')
            exit()



    elif(userinput==13):
        for name in dict3:
            print(name, '-',dict3[name])
            #name is stored in dict3


        j = input("Do you want to continue (YES / NO) - ").lower()
        if j == 'YES' or j == 'yes' or j == 'Yes':
            main()
        else:
            print('===============================================\
                ==================-system end-====================\
                ==========-thank you for using our system-========\
                ================================================')
            exit()
    elif(userinput < 1 or userinput > 13):
        print('please enter valid option')
        #error message
        runAgn= input('\n want to run again? yes/no').lower()
        #it will show in lower case inferior of what the user has made
        if(runAgn=='yes'):
            main()
            #it will restart the program

        else:
            print('''===============================================
                  ==================-system end-====================
                  ==========-thank you for using our system-========
                  ================================================''')
            exit()

# in case the user entered a unknown option
# this is used to complete the defining of the program
#the whole code is defined in main()
main()
```

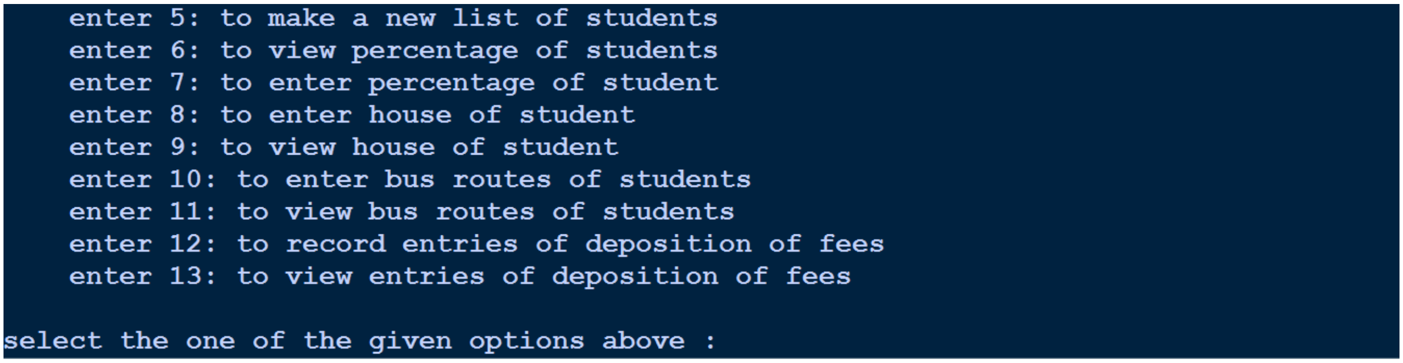
OUTPUT



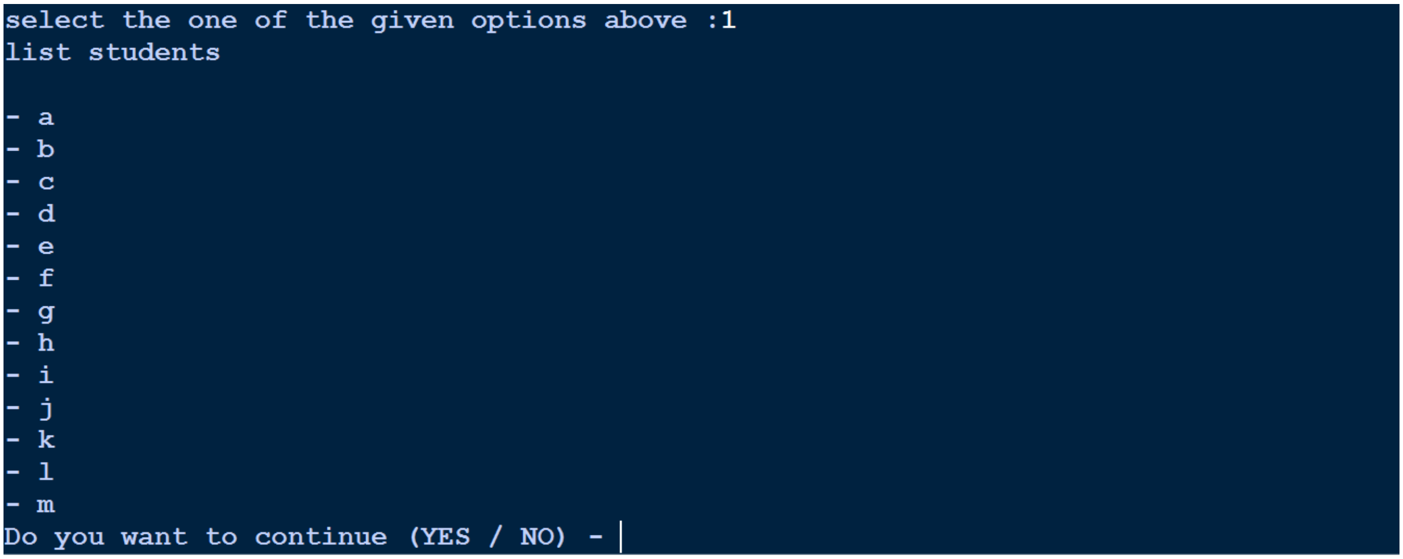
```
======== RESTART: C:\Users\TechFerry\Documents\CS project Hansita\test1.py ========
STUDENT MANAGEMENT SYSTEM ===>
     BENEFITS
===>improves general performance of the students.

===>it would also help in minimal use of papers hence helping in  prevention of pap
er wastage

===>it makes the manual labour of managing these recordes easy

===>it works simply and streamlines all tasks

===>improved communication

===>can be accessed by all parties involved

===>helps to keep track of all students

===>reduction of human labour and workload

===>provides a means to advice students
```



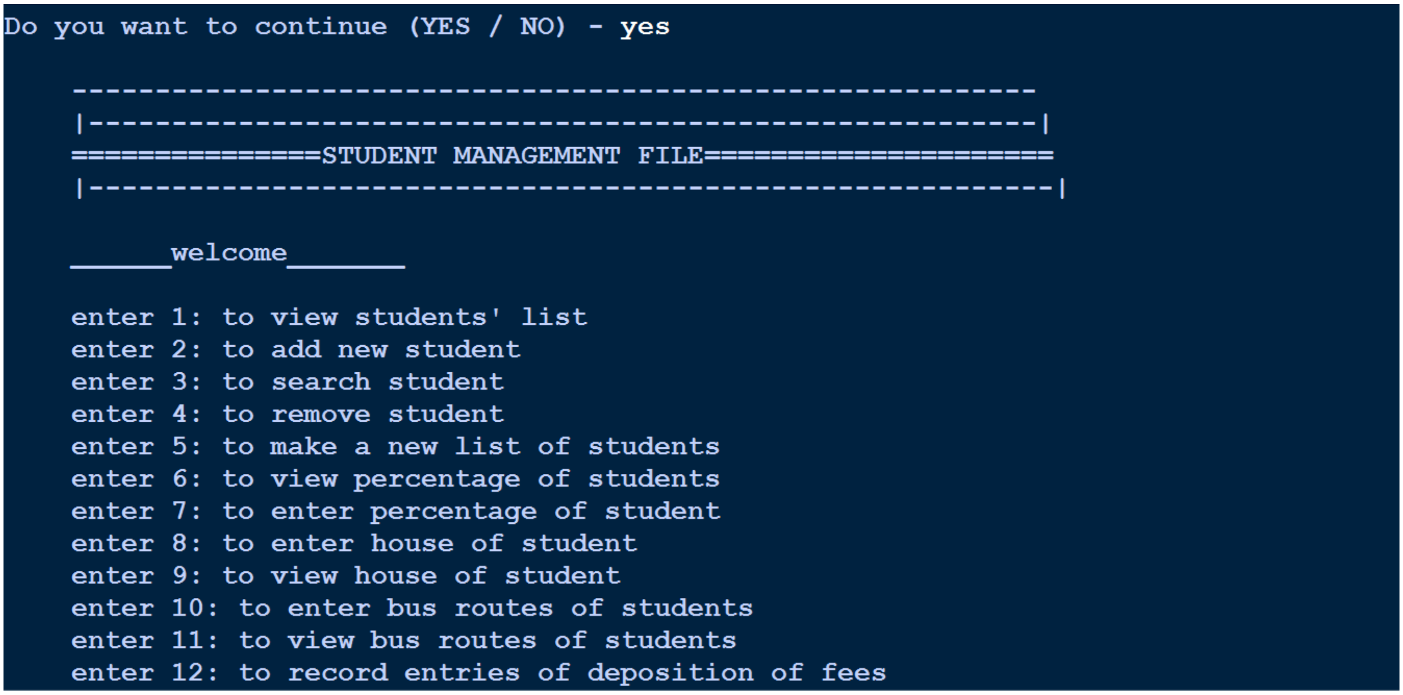
```
===>improves the general comfort of the staff

 \=====-HELPFUL BENFITS OF PYTHON-=========
 ====>it is versatile
\====>easy to use
\===>it is fast to develop
\===> it is open source with a vibrant community
\===>it has all the libraries you can imagine
\===>great prototypes: you can do more by coding less

    -----------------------------------------------------------
    |----------------------------------------------------------|
    ===============STUDENT MANAGEMENT FILE=====================
    |-----------------------------------------------------------|

    ______welcome_______

    enter 1: to view students' list
    enter 2: to add new student
    enter 3: to search student
    enter 4: to remove student
```



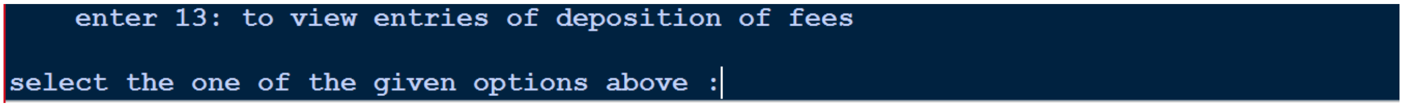
```
    enter 5: to make a new list of students
    enter 6: to view percentage of students
    enter 7: to enter percentage of student
    enter 8: to enter house of student
    enter 9: to view house of student
    enter 10: to enter bus routes of students
    enter 11: to view bus routes of students
    enter 12: to record entries of deposition of fees
    enter 13: to view entries of deposition of fees

select the one of the given options above :
```



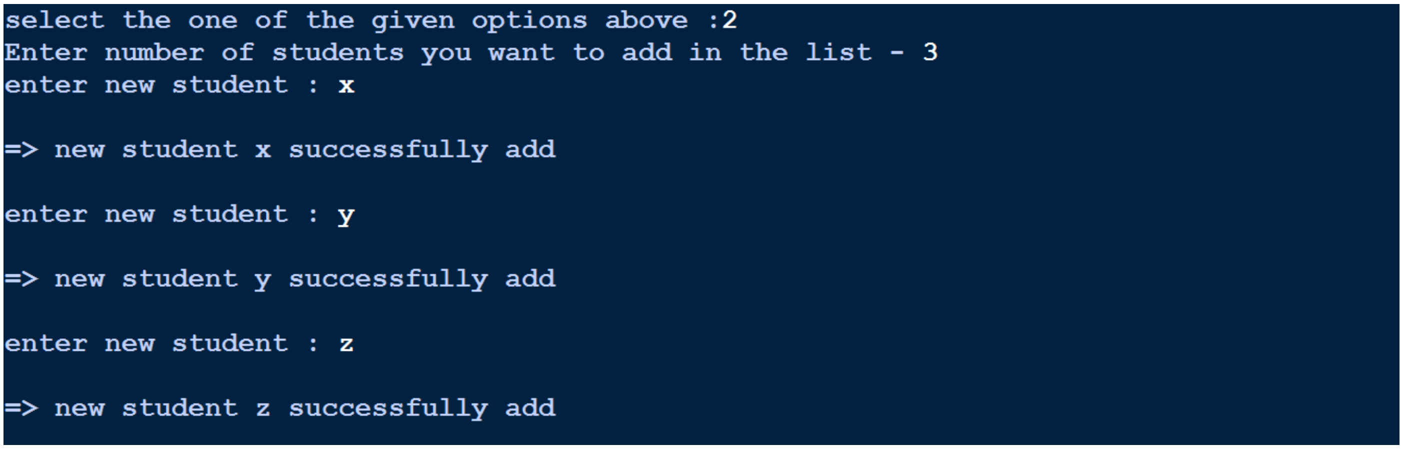
```
select the one of the given options above :1
list students

- a
- b
- c
- d
- e
- f
- g
- h
- i
- j
- k
- l
- m
Do you want to continue (YES / NO) - 
```



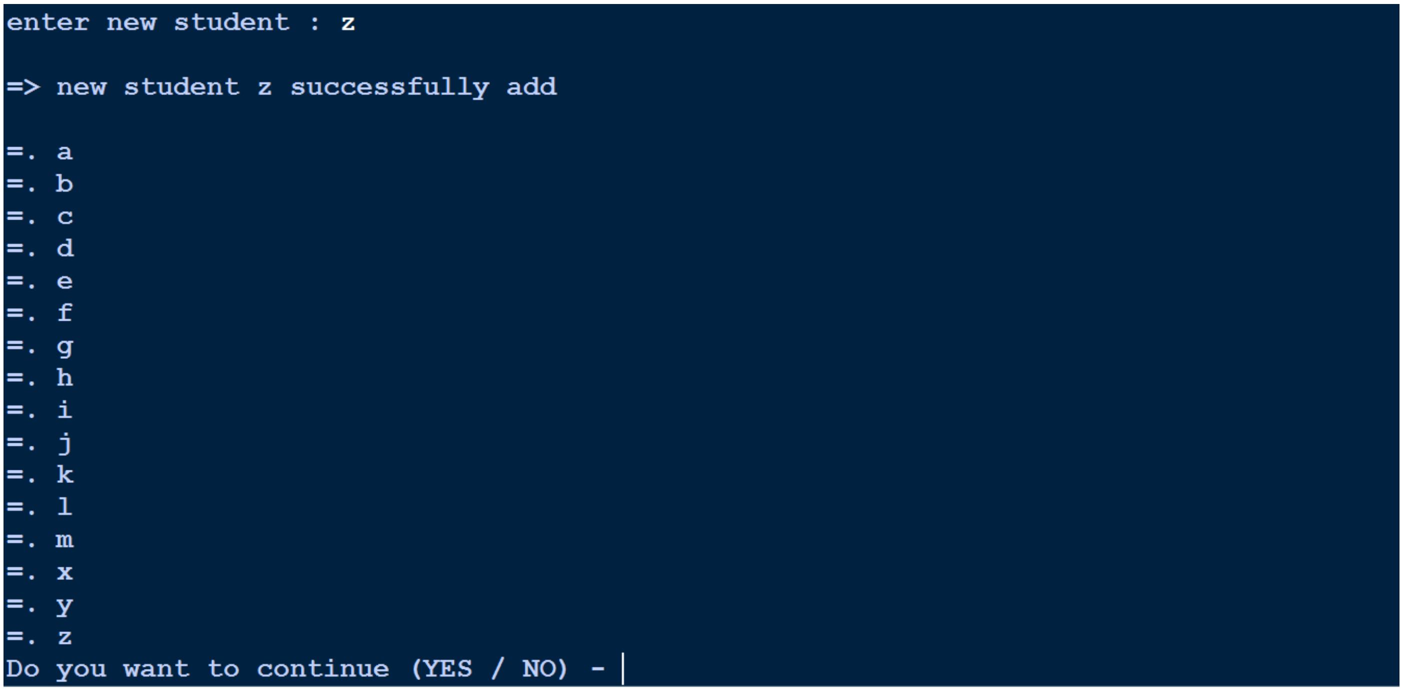
```
Do you want to continue (YES / NO) - yes

    ----------------------------------------------------------
    |---------------------------------------------------------|
    ===============STUDENT MANAGEMENT FILE=====================
    |----------------------------------------------------------|

    ______welcome_______

    enter 1: to view students' list
    enter 2: to add new student
    enter 3: to search student
    enter 4: to remove student
    enter 5: to make a new list of students
    enter 6: to view percentage of students
    enter 7: to enter percentage of student
    enter 8: to enter house of student
    enter 9: to view house of student
    enter 10: to enter bus routes of students
    enter 11: to view bus routes of students
    enter 12: to record entries of deposition of fees
```



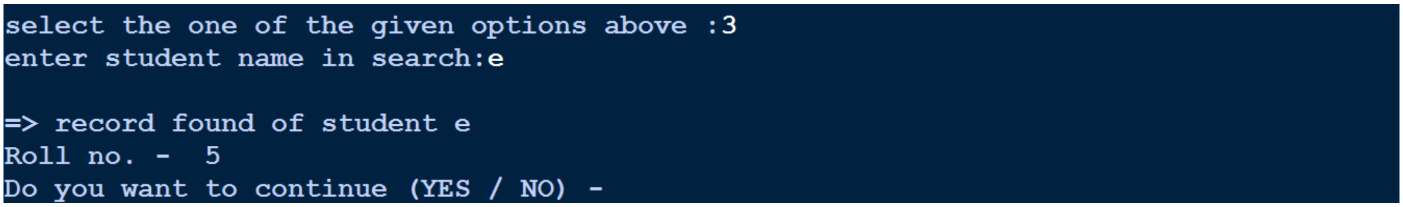
```
    enter 13: to view entries of deposition of fees

select the one of the given options above :
```



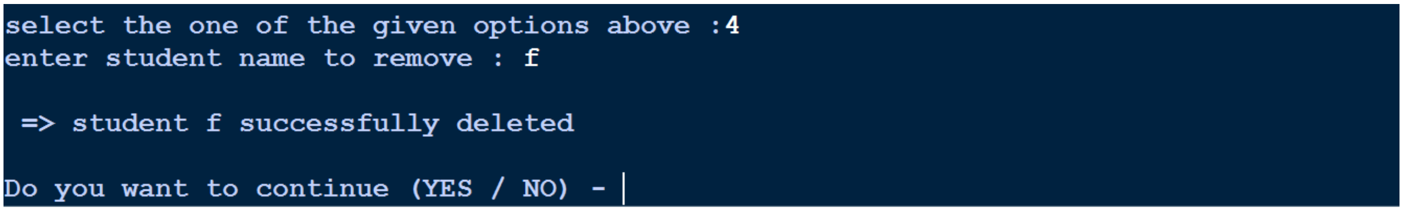
```
select the one of the given options above :2
Enter number of students you want to add in the list - 3
enter new student : x

=> new student x successfully add

enter new student : y

=> new student y successfully add

enter new student : z

=> new student z successfully add
```



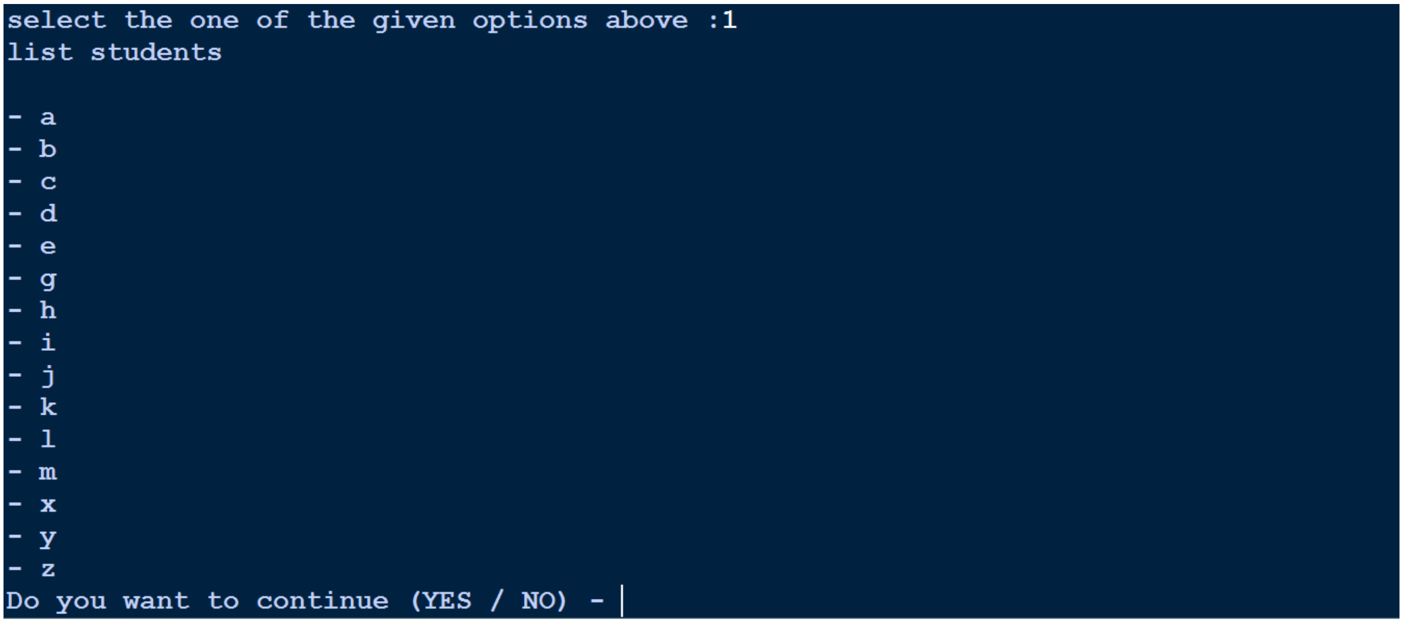
```
enter new student : z

=> new student z successfully add

=. a
=. b
=. c
=. d
=. e
=. f
=. g
=. h
=. i
=. j
=. k
=. l
=. m
=. x
=. y
=. z
Do you want to continue (YES / NO) -
```



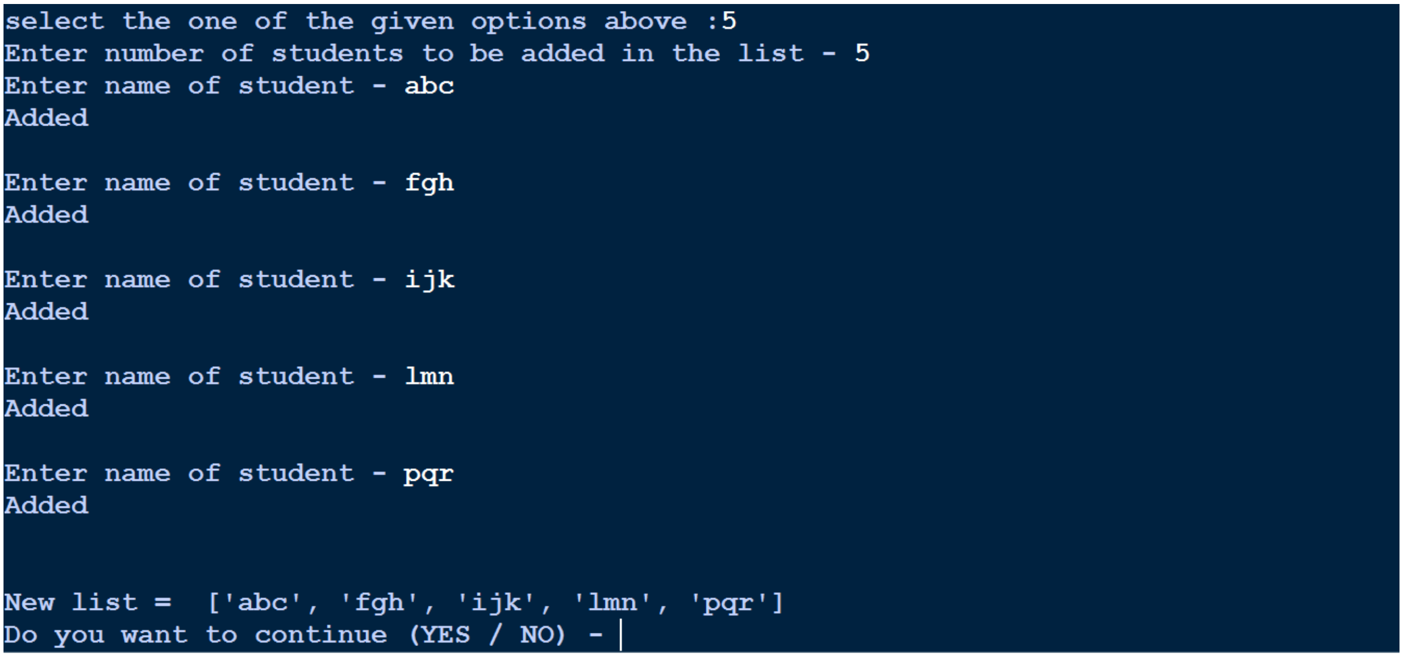
```
select the one of the given options above :3
enter student name in search:e

=> record found of student e
Roll no. -  5
Do you want to continue (YES / NO) -
```



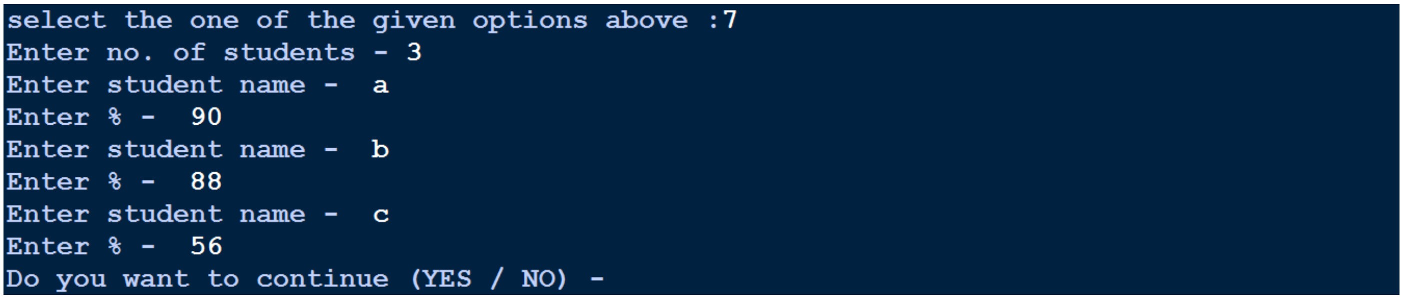
```
select the one of the given options above :4
enter student name to remove : f

 => student f successfully deleted

Do you want to continue (YES / NO) -
```



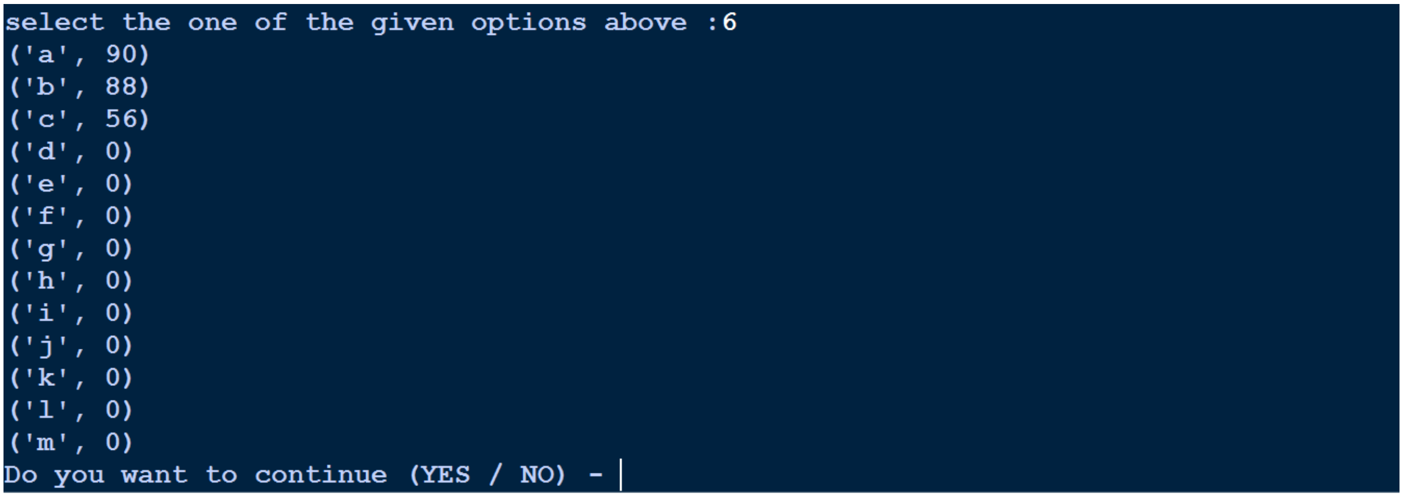
```
select the one of the given options above :1
list students

- a
- b
- c
- d
- e
- g
- h
- i
- j
- k
- l
- m
- x
- y
- z
Do you want to continue (YES / NO) -
```



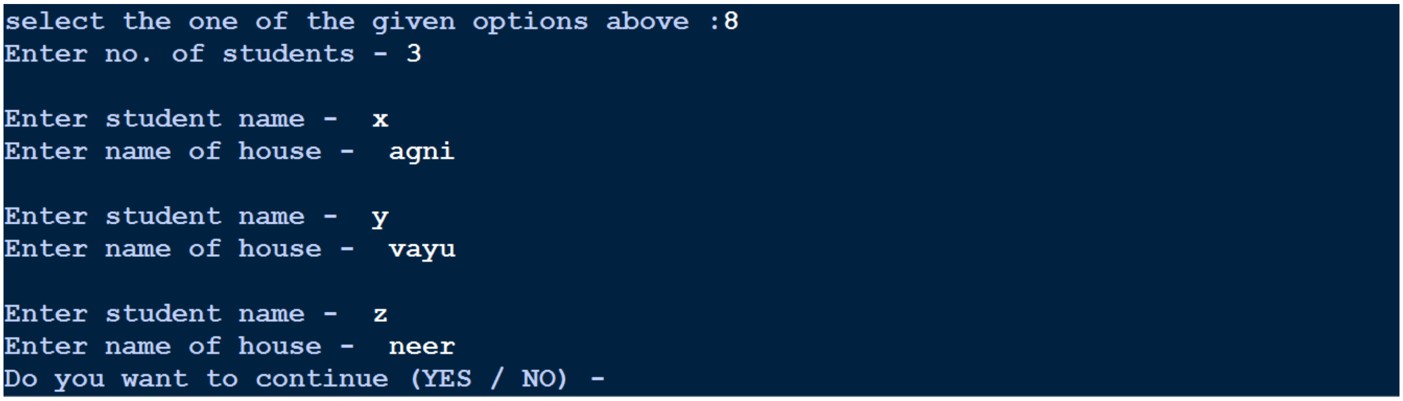
```
select the one of the given options above :5
Enter number of students to be added in the list - 5
Enter name of student - abc
Added

Enter name of student - fgh
Added

Enter name of student - ijk
Added

Enter name of student - lmn
Added

Enter name of student - pqr
Added


New list =  ['abc', 'fgh', 'ijk', 'lmn', 'pqr']
Do you want to continue (YES / NO) - 
```



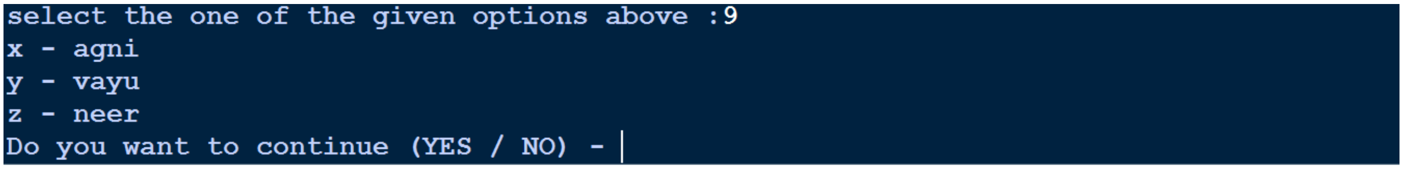
```
select the one of the given options above :7
Enter no. of students - 3
Enter student name -  a
Enter % -  90
Enter student name -  b
Enter % -  88
Enter student name -  c
Enter % -  56
Do you want to continue (YES / NO) -
```



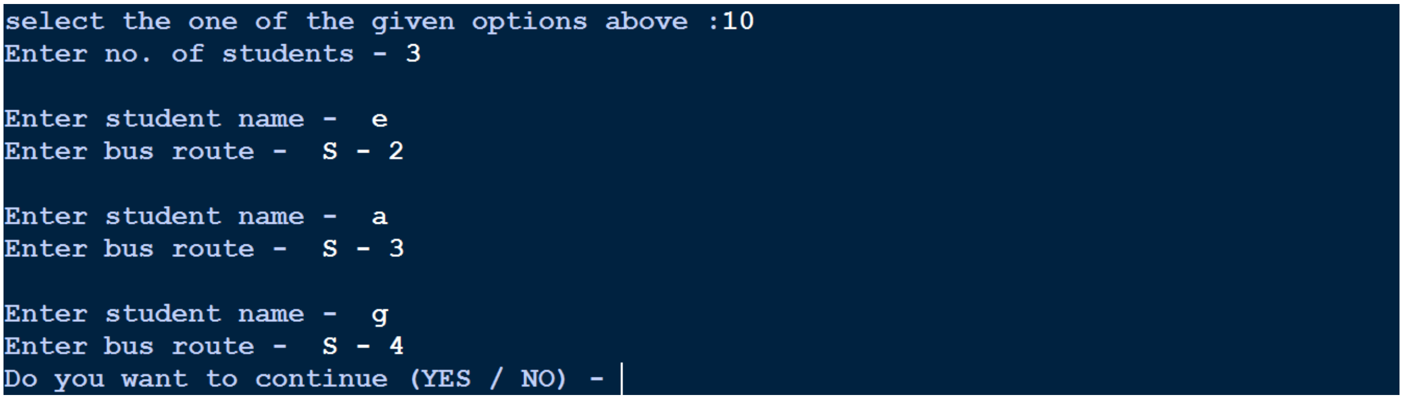
```
select the one of the given options above :6
('a', 90)
('b', 88)
('c', 56)
('d', 0)
('e', 0)
('f', 0)
('g', 0)
('h', 0)
('i', 0)
('j', 0)
('k', 0)
('l', 0)
('m', 0)
Do you want to continue (YES / NO) - 
```



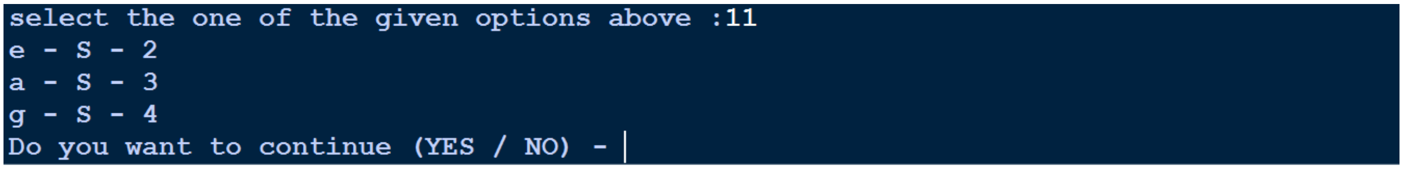
```
select the one of the given options above :8
Enter no. of students - 3

Enter student name -  x
Enter name of house -  agni

Enter student name -  y
Enter name of house -  vayu

Enter student name -  z
Enter name of house -  neer
Do you want to continue (YES / NO) -
```



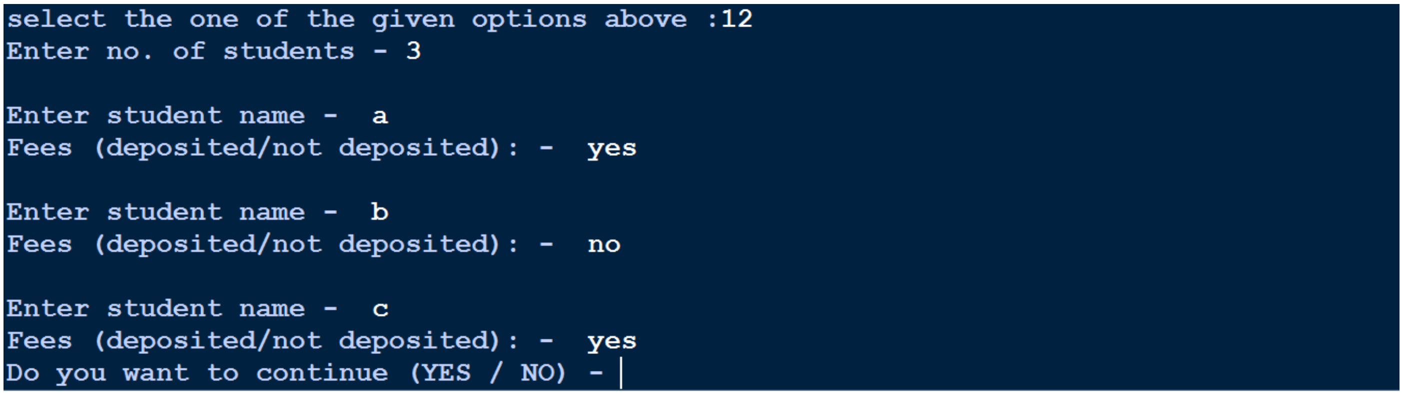
```
select the one of the given options above :9
x - agni
y - vayu
z - neer
Do you want to continue (YES / NO) -
```



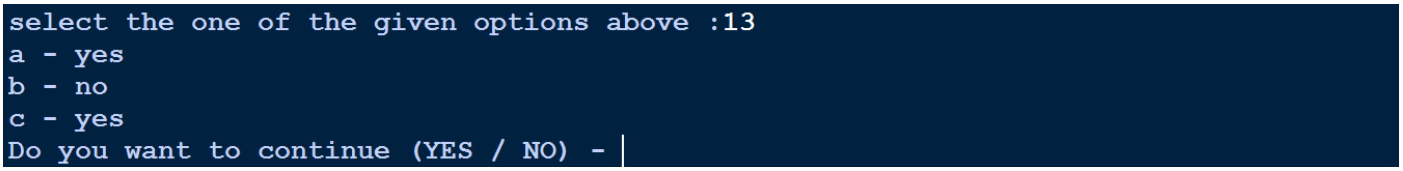
```
select the one of the given options above :10
Enter no. of students - 3

Enter student name -  e
Enter bus route -  S - 2

Enter student name -  a
Enter bus route -  S - 3

Enter student name -  g
Enter bus route -  S - 4
Do you want to continue (YES / NO) -
```



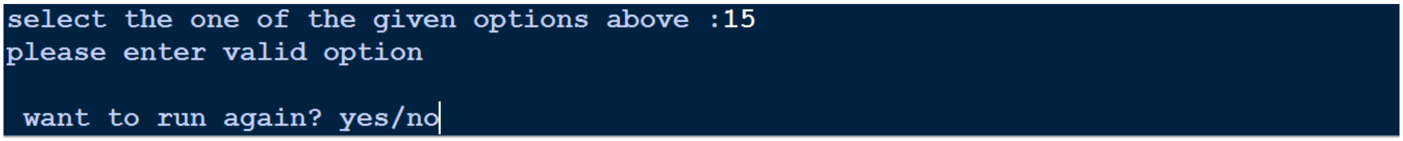
```
select the one of the given options above :11
e - S - 2
a - S - 3
g - S - 4
Do you want to continue (YES / NO) -
```



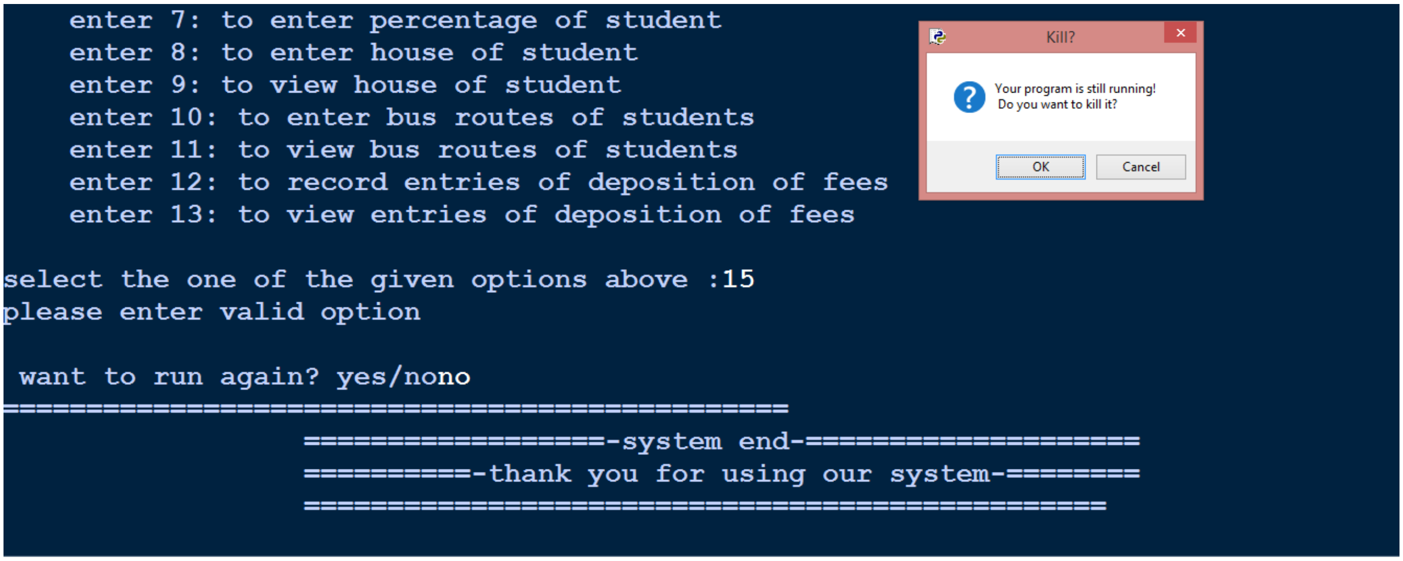
```
select the one of the given options above :12
Enter no. of students - 3

Enter student name -  a
Fees (deposited/not deposited): -  yes

Enter student name -  b
Fees (deposited/not deposited): -  no

Enter student name -  c
Fees (deposited/not deposited): -  yes
Do you want to continue (YES / NO) -
```



```
select the one of the given options above :13
a - yes
b - no
c - yes
Do you want to continue (YES / NO) -
```



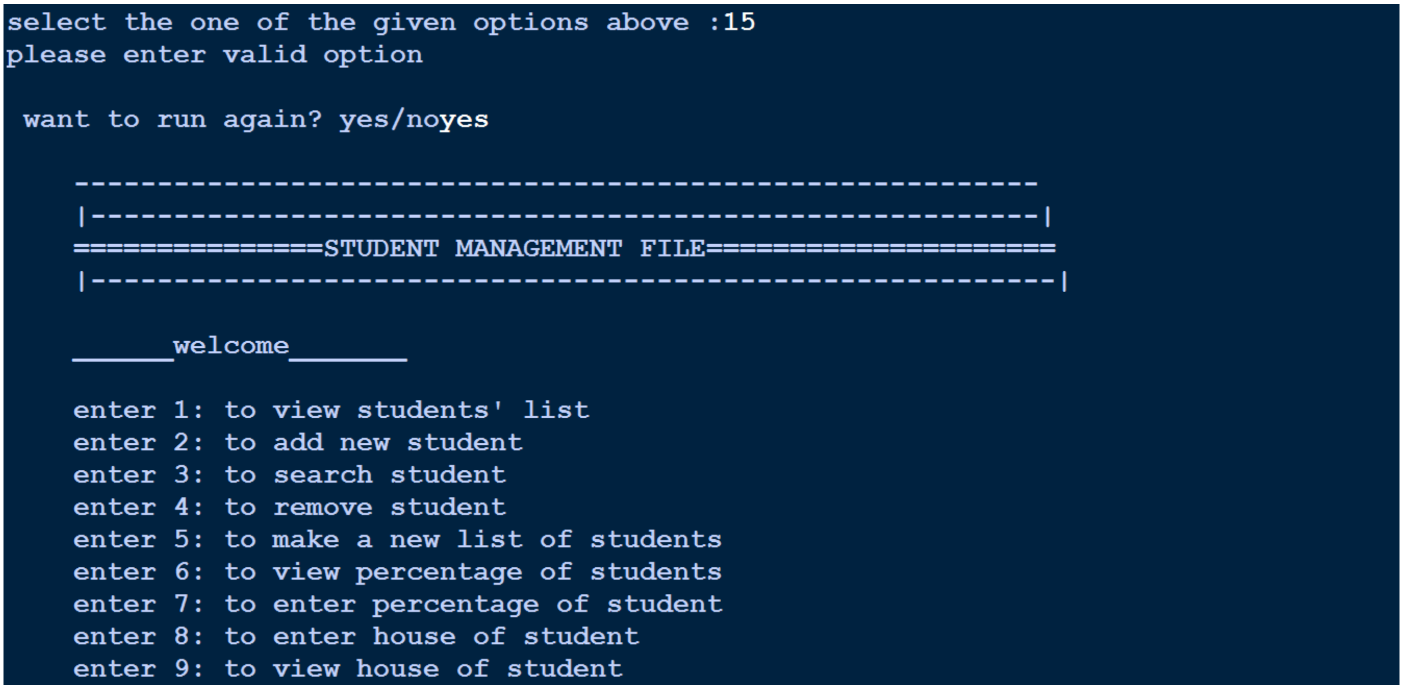
```
select the one of the given options above :15
please enter valid option

 want to run again? yes/no
```



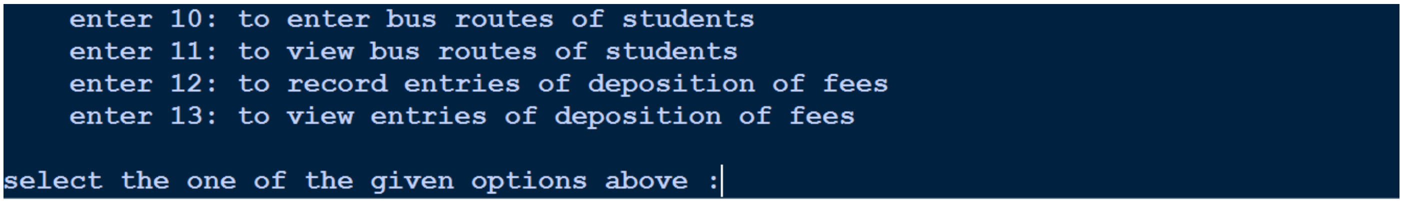
```
    enter 7: to enter percentage of student
    enter 8: to enter house of student
    enter 9: to view house of student
    enter 10: to enter bus routes of students
    enter 11: to view bus routes of students
    enter 12: to record entries of deposition of fees
    enter 13: to view entries of deposition of fees

select the one of the given options above :15
please enter valid option

 want to run again? yes/nono
================================================
                  ==================-system end-====================
                  ==========-thank you for using our system-========
                  ==================================================
```

Kill?

Your program is still running!
Do you want to kill it?



```
select the one of the given options above :15
please enter valid option

 want to run again? yes/noyes

    ----------------------------------------------------------
    |-----------------------------------------------------------|
    ===============STUDENT MANAGEMENT FILE=====================
    |------------------------------------------------------------|

    ______welcome_______

    enter 1: to view students' list
    enter 2: to add new student
    enter 3: to search student
    enter 4: to remove student
    enter 5: to make a new list of students
    enter 6: to view percentage of students
    enter 7: to enter percentage of student
    enter 8: to enter house of student
    enter 9: to view house of student
```



```
    enter 10: to enter bus routes of students
    enter 11: to view bus routes of students
    enter 12: to record entries of deposition of fees
    enter 13: to view entries of deposition of fees

select the one of the given options above :
```

Bibliography

- https://www.python.org/
- https://www.w3schools.com/
- https://www.geeksforgeeks.org/python-programming-language/
- https://www.programiz.com/python-programming
- https://docs.python.org/
- https://urvashisinghal.blogspot.com/p/class-xi-study-material.html

This Photo by Unknown Author is licensed under CC BY-NC-ND