

## THEORY:-

### # LIST:-

→ The List is a most versatile datatype available in python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type. creating a list is as simple as putting different comma-separated values between square brackets

### For Example:-

```
List1 = ['physics', 'chemistry', 1997, 2000];
```

```
List2 = [2, 2, 3, 4, 5]
```

```
List3 = ["a", "b", "c", "d"]
```

### Accessing Values in a List:-

To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at the index.

```
List1 = ['physics', 'chemistry', 1997, 2000]
```

```
print("List1[0]:", List1[0])
```

```
print("List1[1:2]:", List1[1:2])
```

### # OUTPUT:-

```
List1[0]: physics.
```

```
List1[1:2]: ['chemistry', 1997]
```

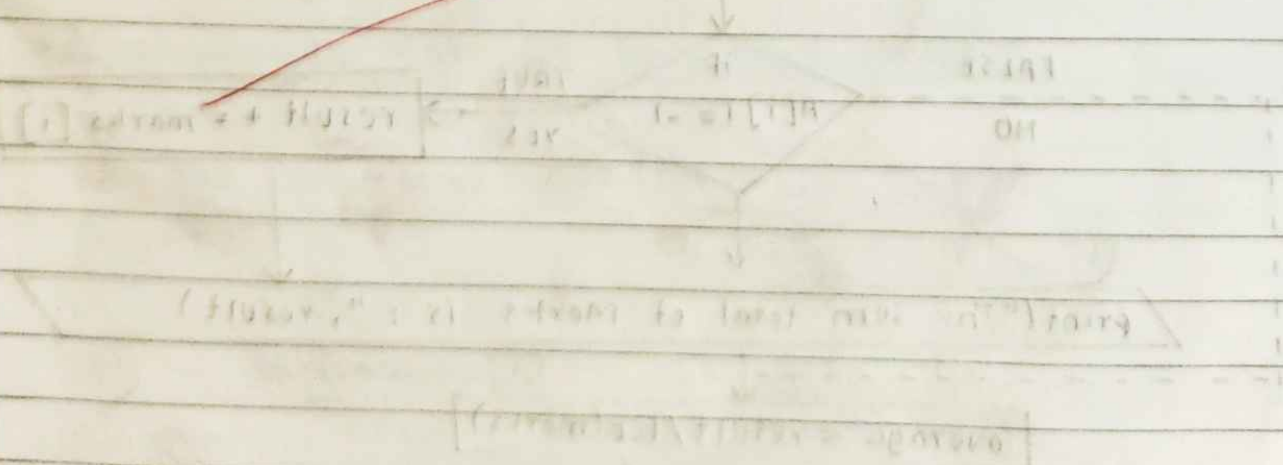
⇒ (1) Pseudocode for def Accept\_marks():

1. Input the total number of students in FDS
2. for i in range(n):
  - m = int(input("Enter marks scored by each student in FDS (out of 30) : \t"))
3. marks.append(m)
4. print("Marks scored by each student in FDS (list) : \t", marks)
5. Stop.

⇒ (2) Pseudocode for def Average\_marks():

1. Initialize result = 0
2. For i in range(len(marks)):
  - (begin)
  - if marks[i] != -1:
    - result = result + marks[i]
  - (End)
- print("The Sum total of Marks is : ", result)
2. store average = result / len(marks)
4. print("The average of total marks : ", average)
5. Stop.

print average marks



Function def

⇒ (3) Pseudocode For highest\_marks\_and\_lowest\_marks():

1. Initialize max = -1
2. Initialize min = 31
3. For i in range(1, len(marks)): (do)

(begin)

if (max < marks[i]) then

max = marks[i]

max\_ind = i

if (min > marks[i] and marks[i] != -1) then

min = marks[i]

min\_ind = i

(end)

4. print("Highest mark scored %d by student %d" % (max, max\_ind + 1))
5. print("Lowest mark scored %d by student %d" % (min, min\_ind + 1))

⇒ (4) Pseudocode For absent\_students():

1. Initialize counter = 0
  2. For i in range(len(marks)) then
- (begin) if mark[i] == -1 then  
counter = counter + 1

(end)

3. print("The Number of students Absent:", counter)

⇒ (5) Pseudocode for def high\_frequency():

1. Initialize Freq\_count = 0
2. For in range(len(marks)) (do)

(begin)

3. Initialize count = 0

4. If (marks[i] != -1) then

5. For j in range(len(marks)):

(begin)

if (marks[i] == marks[j]) then  
count += 1

(end)

6. if (Freq < count) then  
store Marks = marks[i]  
store freq = count.

(end)

7. print ("\n Marks with highest Frequency is %d (%d)"  
% (Marks, Frequency))

⇒ (6) PSEUDO-CODE for function def main():

1. Start

2. Create a list marks = []

3. Using while loop

while True:

4. print ("=====")

print Accept FDS Marks.

print Average Score of the class.

print Highest score and lowest score of the class.

print Count of students who were absent for the test

print Display marks with Highest Frequency.

print Exit.

5. ch = int(input("Enter your choice :"))

6. if (ch > 5) then

print END OF PROGRAM

quit()

elif (ch == 1) then

call function Accept\_marks()

elif (ch == 2) then

call function Average\_marks()

elif (ch == 3) then

call function highest\_marks\_and\_lowest\_marks()

elif (ch == 4) then  
Call Function absent\_students()

elif (ch == 5) then

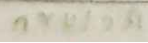
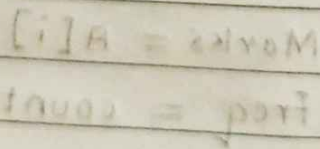
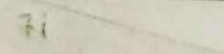
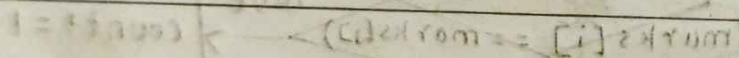
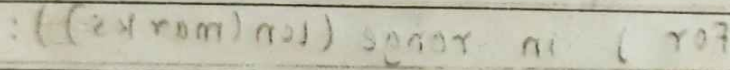
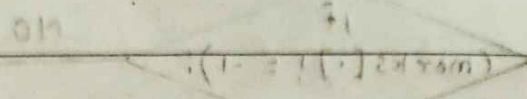
Call Function high\_Frequency()

else then

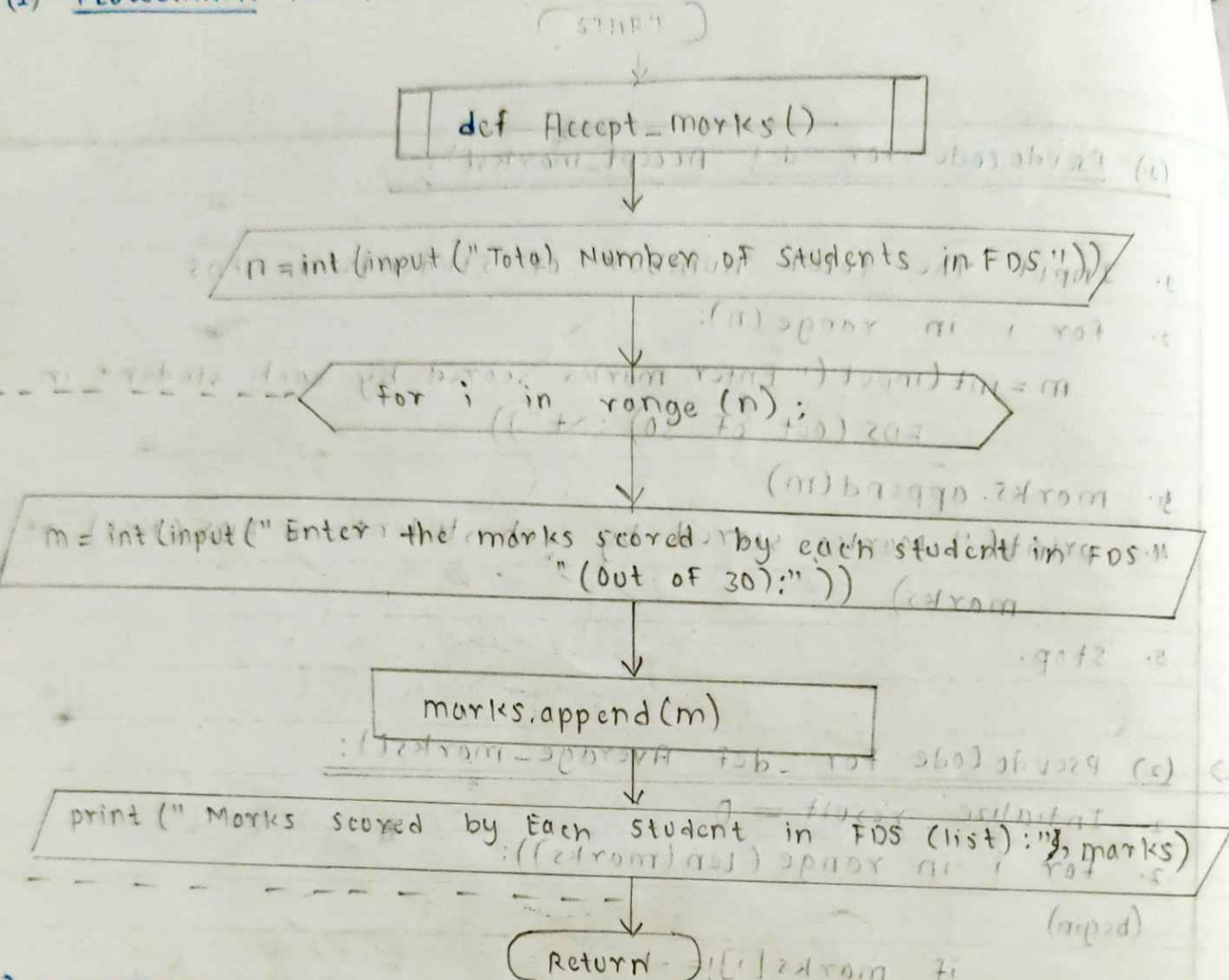
print Wrong choice Entered !! Try Again.

Call Main() Function

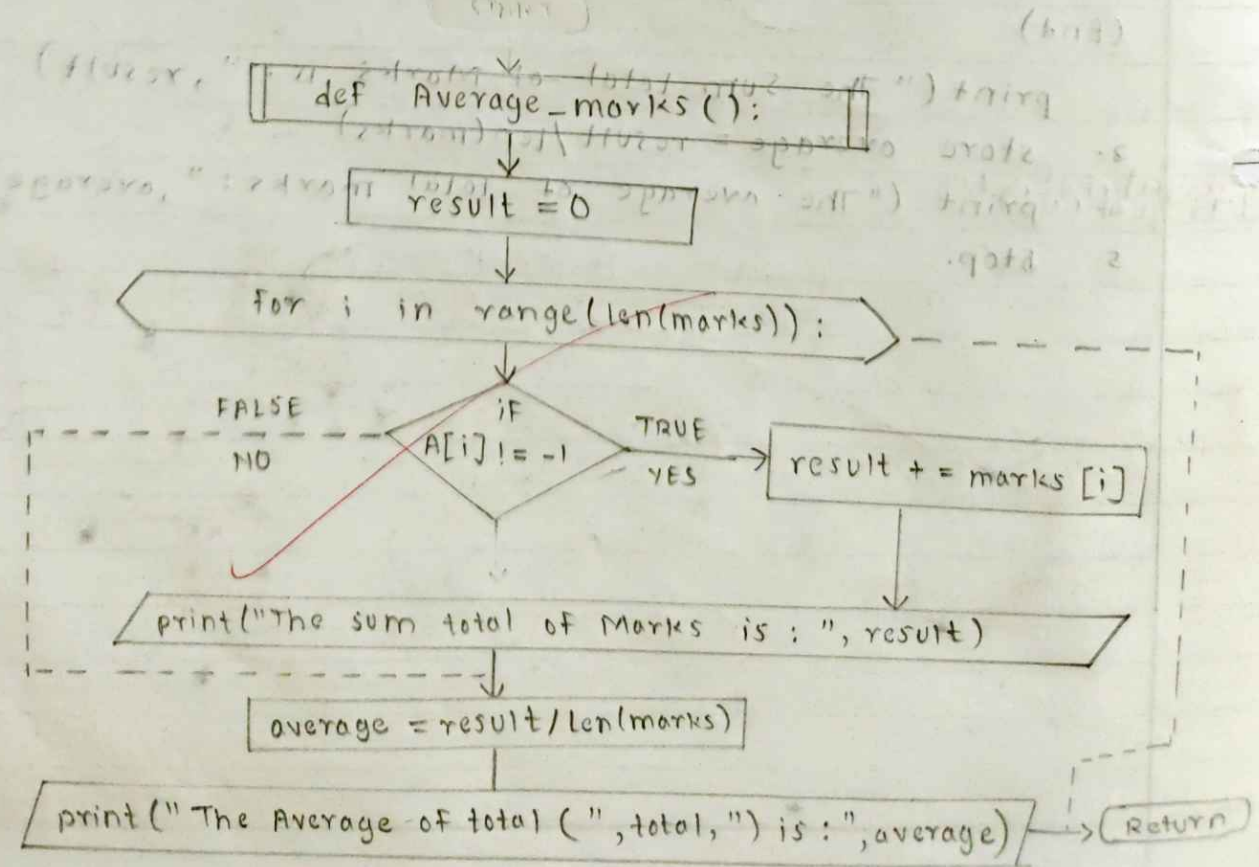
7. stop.



(2) FLOWCHART: For Function `def Accept_marks():`



(2) FLOWCHART:- For Function `def Average_marks():`



(3) ~~FOR~~ FLOWCHART: FOR Function highest\_marks\_and\_lowest\_marks

def highest\_marks\_and\_lowest\_marks():

max = -1  
min = 31

for i in range(0, len(marks)):

if (max < marks[i]):

max = marks[i]  
max\_ind = i

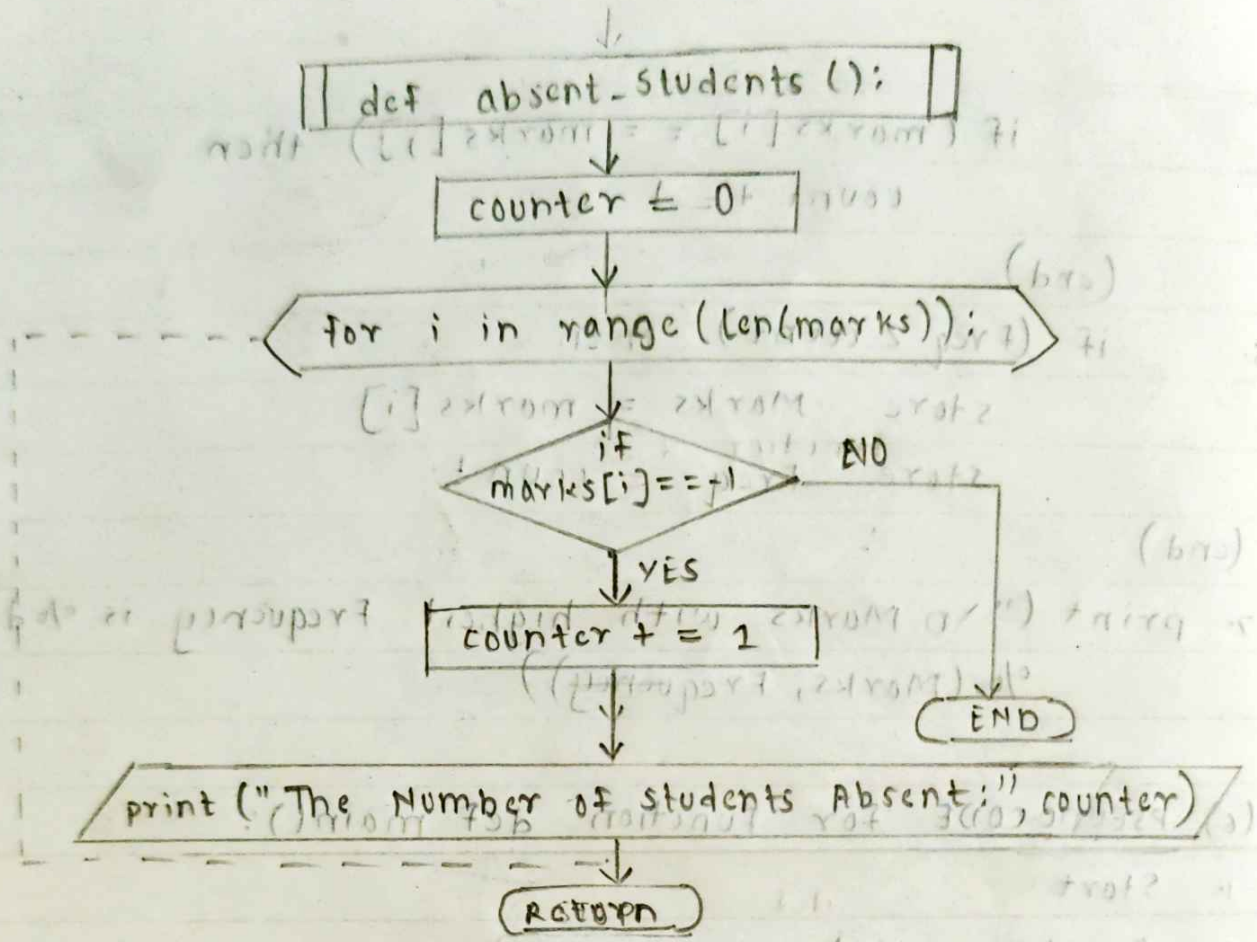
if (min > marks[i] and marks[i] != -1):

min = marks[i]  
min\_ind = i

print ("Highest Mark score of class is %d scored by student %d" % (max, max\_ind+1))  
print ("Lowest Mark score of class is %d scored by student %d" % (min, min\_ind+1))

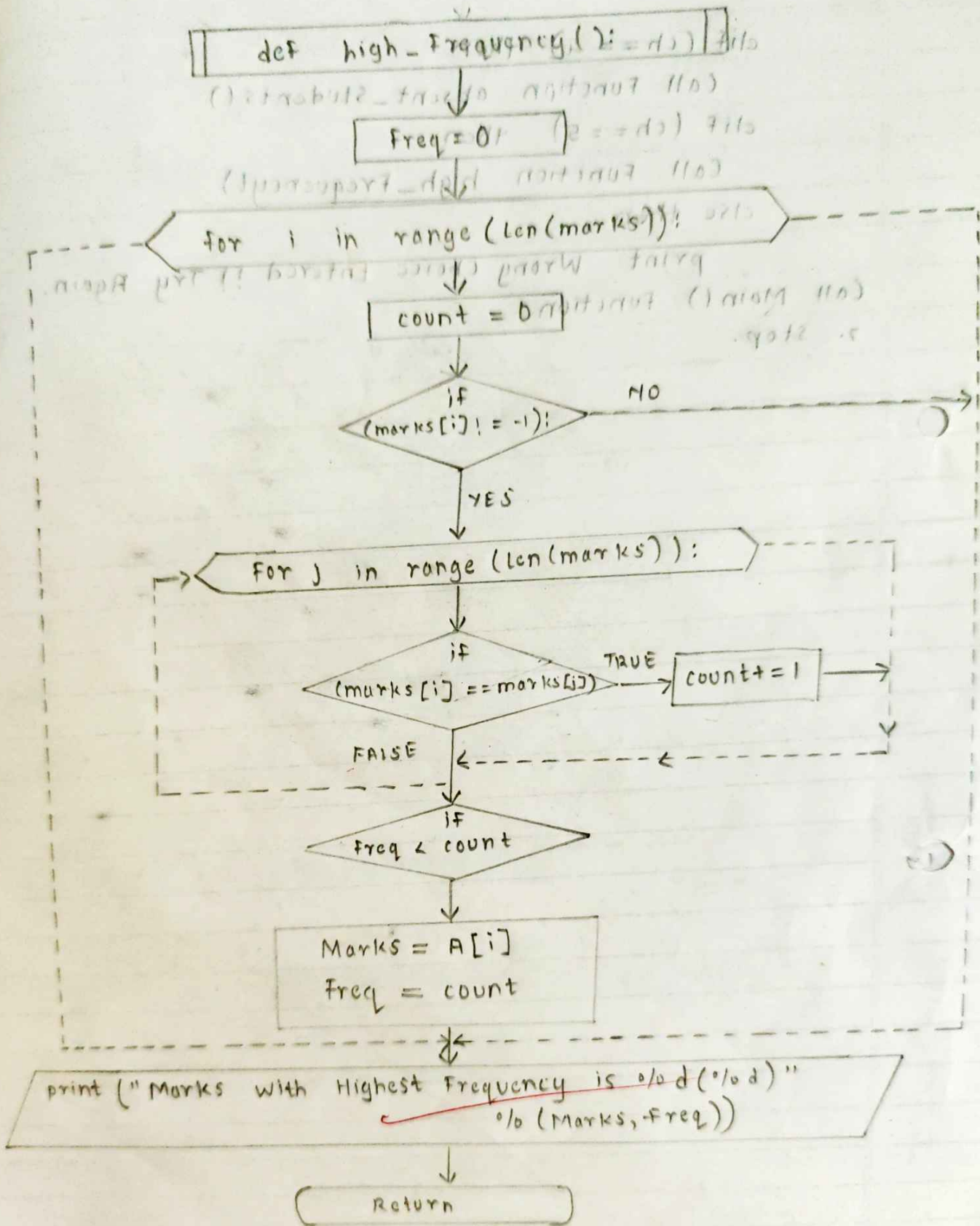
Return

(4) FLOWCHART - For Function `absent_students()`:





FLOWCHART! - for FUNCTION def high-frequency():



Flowchart :-

For Function def main():

marks = []

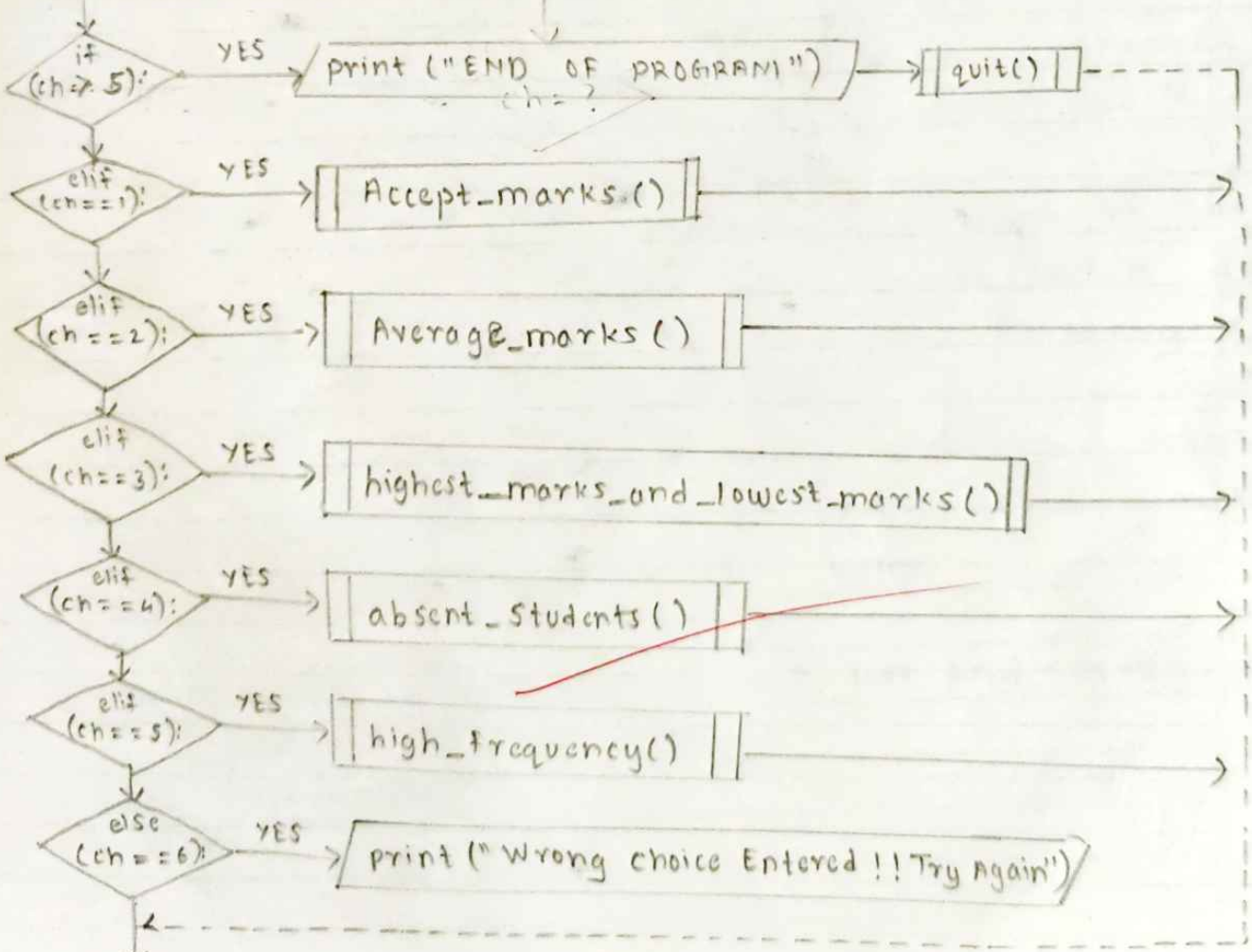
while True:

```

print("===== ")
print(" 1: Accept FDS Marks")
print(" 2: Average Score of Class")
print(" 3: Highest score and lowest core of class")
print(" 4: Count of students who were absent for the test")
print(" 5: Display Mark with Highest Frequency")
print(" 6: EXIT")
print("===== ")

```

ch = int(input("Enter your choice : "))



Stop

(Return)

# ASSIGNMENT: NO: 2: (TWO): -

0. QUESTIONS:-
1. What is static and dynamic memory allocation?

ANS.

Static Memory Allocation:-

When the allocation of memory performs at the compile time, then it is known as static memory. In this, the memory is allocated for variables by the compiler. It saves running time as it is fast. static memory allocation is preferred in an array. It allots the memory from stack.

Dynamic Memory Allocation:-

When the memory allocation is done at the execution or runtime, then it is called dynamic memory allocation. In this, the memory is allocated to variables during execution. It is slower than static memory allocation. Dynamic memory allocation is preferred in the linklist. It allots memory from the heap.

2. Explain difference between list and array in python with an example

ANS.

DIFFERENCE:-

SR.NO.:-

0 LIST:-

NO.:-

1D ARRAY:-

1. List is used to collect items that usually consists of elements of multiple data - types.

2. List cannot manage Arithmetic Operations.

3. It consists of elements that belong to the different data types.

4. List is more flexible as it allows easy modification of data.

5. It consumes a larger memory.

6. In a list, the complete list can be accessed without any specific looping.

7. It favours a shorter sequence of data.

→ EXAMPLE:-

```
sample_list = [1, "yash", ['a', 'e']]
print(type(sample_list))
print(sample_list)
```

OUTPUT:-

```
<class 'list'>
[1, 'yash', ['a', 'e']]
```

~~Paul~~

2. An Array is also a vital component that collects several items of the same datatype.

3. Array can manage Arithmetic Operations.

4. It consists of elements that belong to the same data-types.

5. List is less flexible, as it doesn't allow easy modification of data.

6. It consumes less memory than a list.

7. In an array, a loop is mandatory to access the components of the array.

8. It favours a longer sequence of data.

→ EXAMPLE:-

```
import array as arr
a = arr.array('i', [1, 2, 3])
print(type(a))
for i in a:
```

```
print(i, end = " ")
```

OUTPUT:-

```
< class 'array.array' >
1 2 3
```