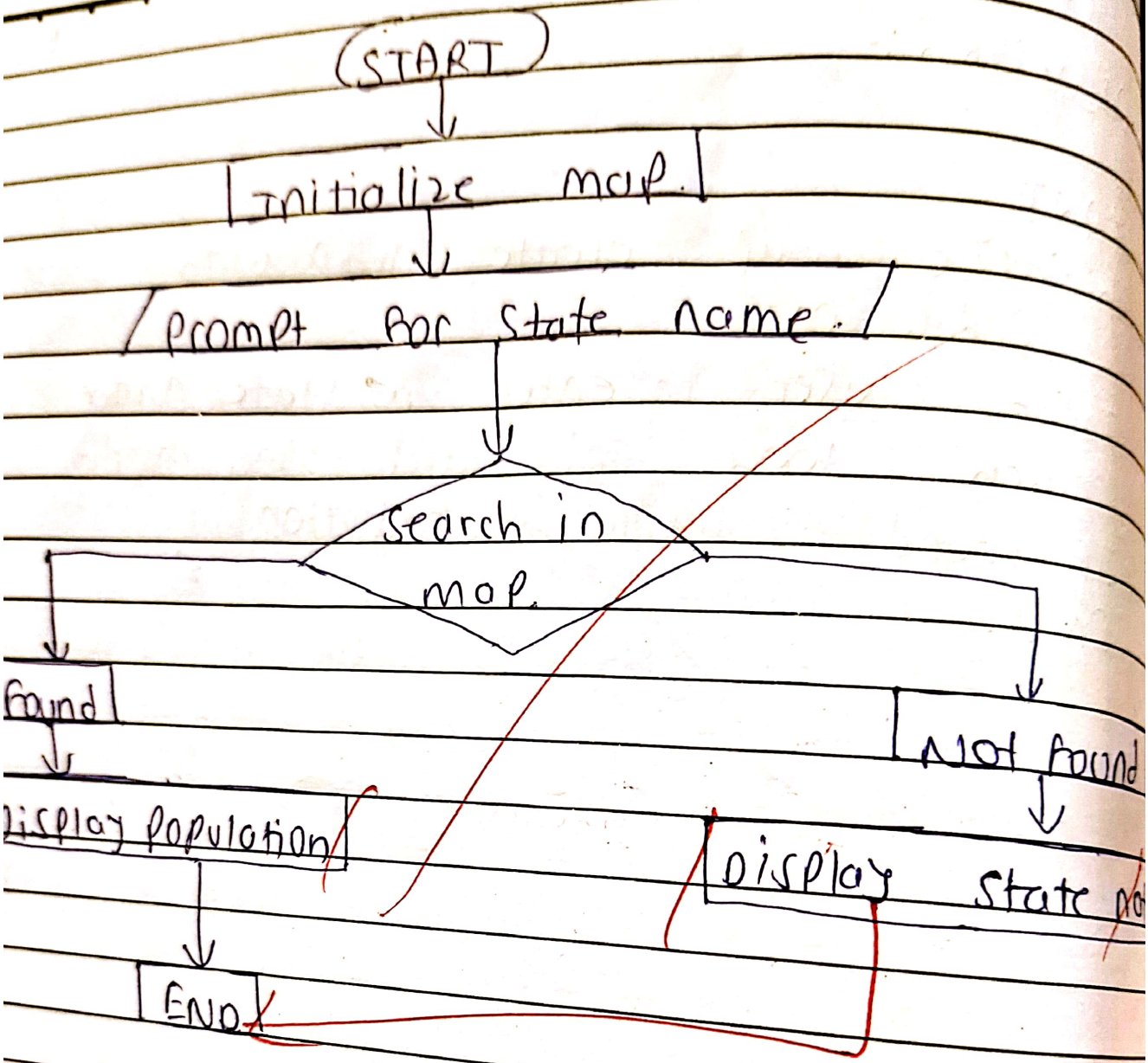
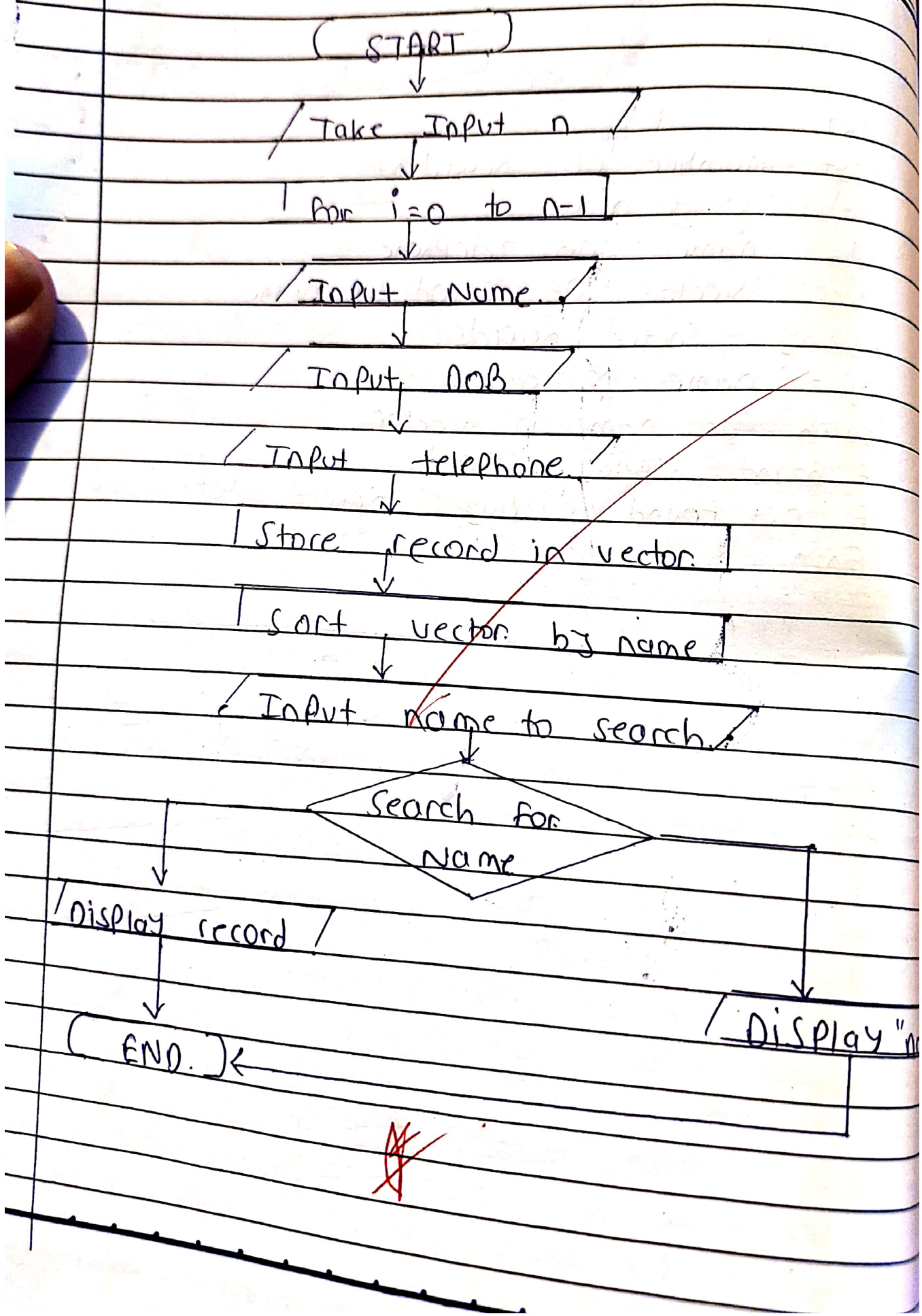
GURU GOBIND SINGH FOUND
POLYTECHNIC / DEGREE

Date :



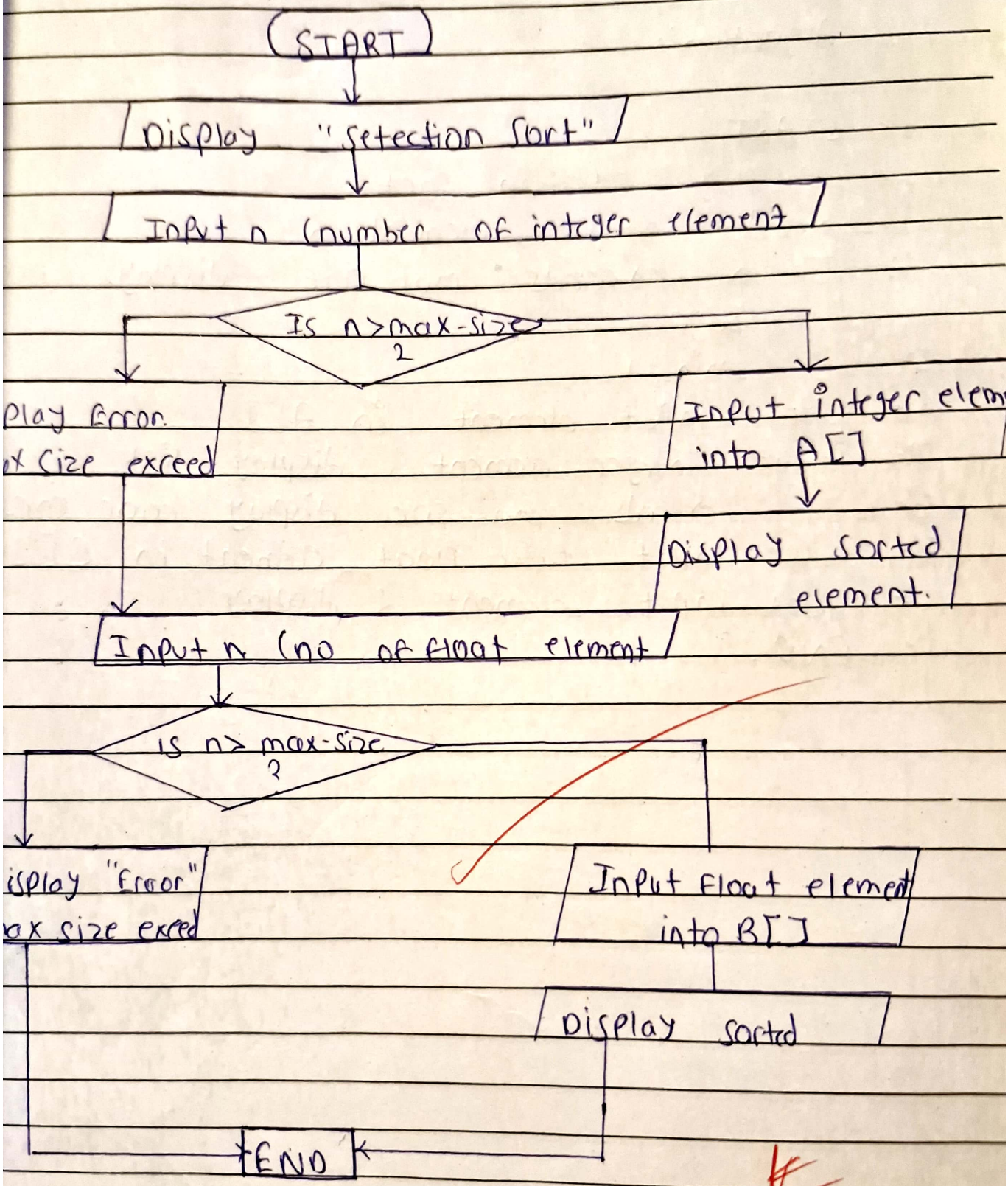
Algorithm :-

START.
Initialize map : create map with state name & population
Input : user to enter a state name.
search : look for state in map.
If found display population.
END.



Algorithm :-

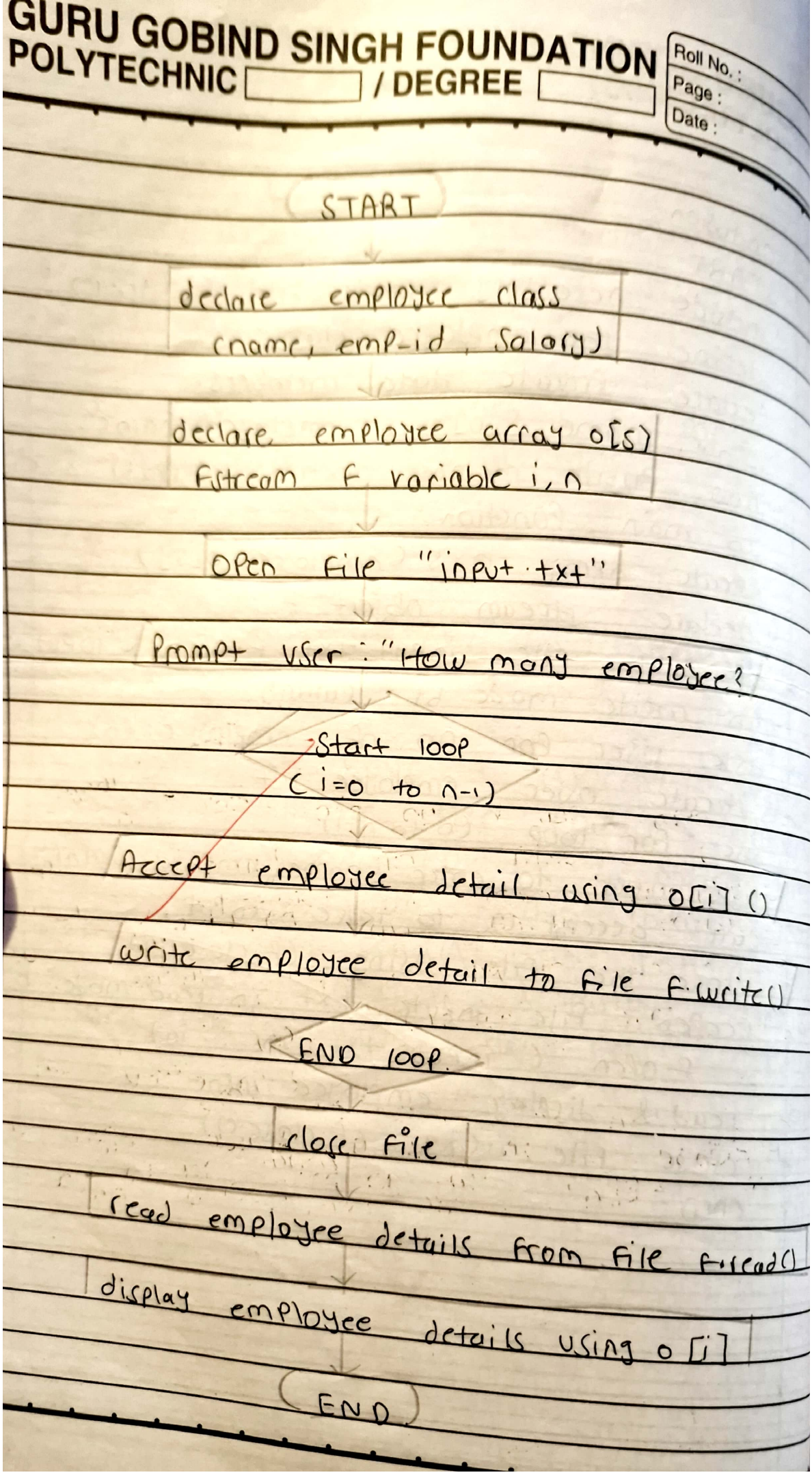
START.
Input number OF Record n)
for i = 0 to n-1
Input name, DOB, TelePhone
Stores vector OF recod by nome.
Display Sorted Records.
Input name to search.
Search for name in vector.
If found, display record.
If not found, display "Record not found"
END.



Algorithm :-

1. START
2. Display selection sort
3. Read integer elements n.
4. If n exceeds max size display error message.
5. Set min = 1
6. Sort input element in A[]
7. input integer element & display list.
8. IF exceeds max-size display error message.
9. Else input take float element in B[]
10. sort input element & display.
11. END.

Roll No. :
Page :
Date :

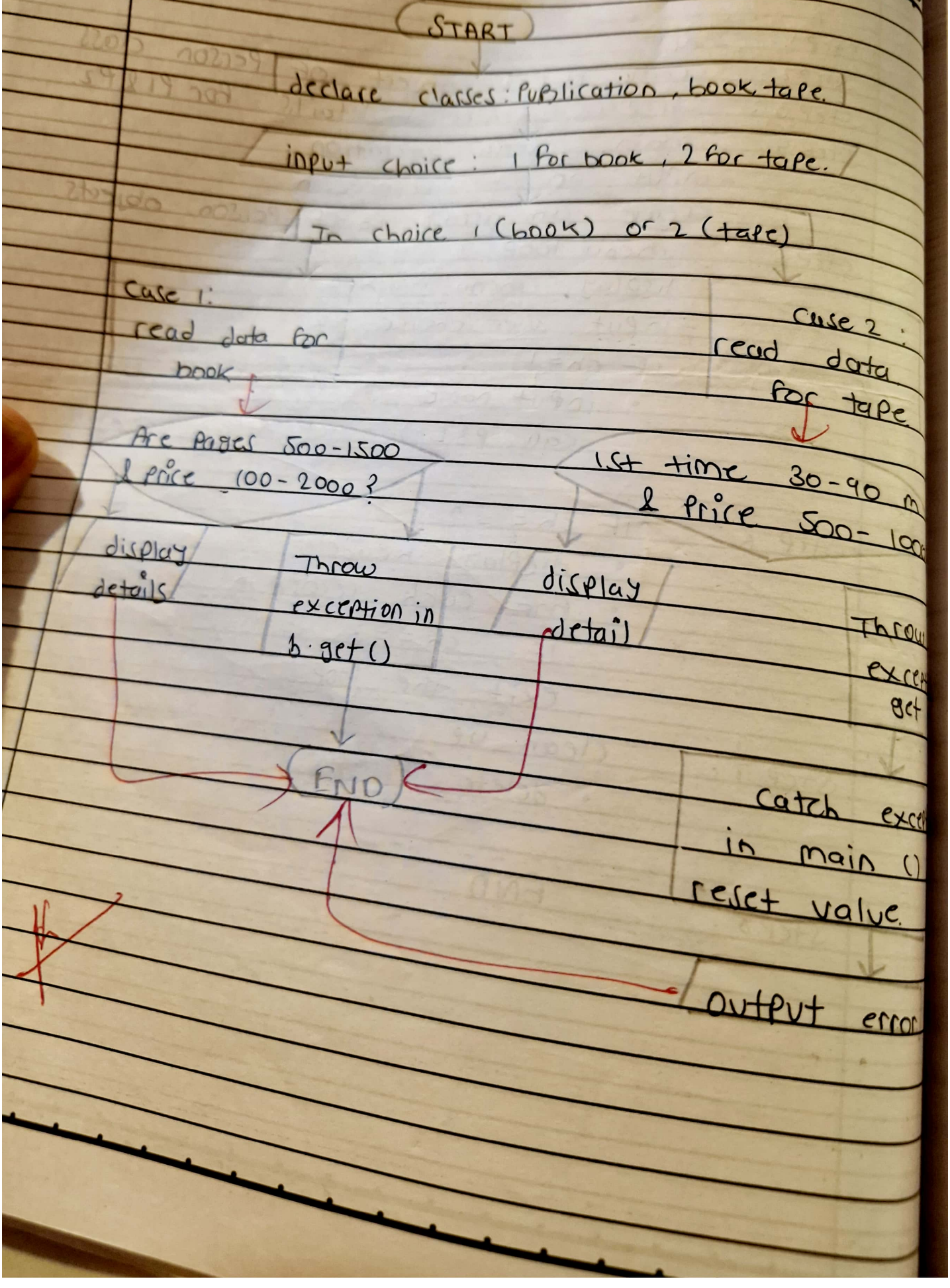


conclusion:

START
include necessary header file (iostream & fstream)
define the employee class:
- declare private data members:
- char name [20] for employee name.
- Two public member function accept() & display()
In main function.
- create array 0-5 (employee o[5])
- declare stream object f.
open the file input.txt f.open ("input.txt")
in write mode by calling
Ask user for no of employee (n)
itrate over employee list.
use for loop (0 to n-1)
ask user to enter (name, emp-id, salary)
call accept () to take input.
close the file using (f.close())
reopen file input.txt in read mode by calling
f.open ("input.txt, ios::in)
read & display employee info-
close file using (f.close())
END.

GURU GOBIND SINGH FOUNDATION
POLYTECHNIC / DEGREE

Roll No. :
Page :
Date :



Algorithm :

step 1: start
step 2: create P1 & P2 object of person class
step 3: display header & details for P1 & P2.
step 4: main menu operation.
- input no of records n
- create an array P3 of person objects.
step 5: menu loop.
- display menu option.
- input user choice ch.
if ch = 1
- input name
- call P3[x] get data (name)

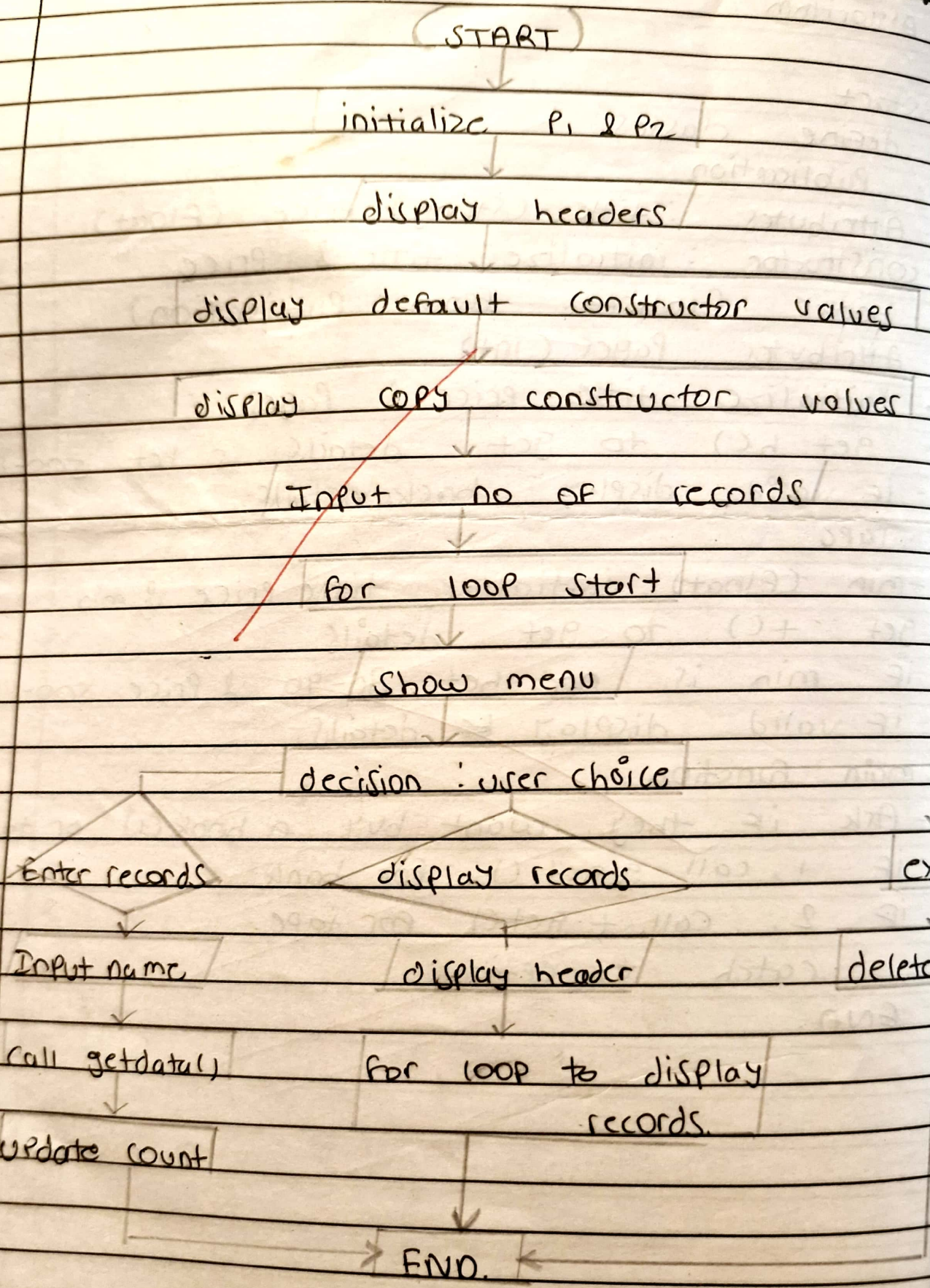
step 6: if ch = = 2
- display headers.
- for each record in P3, call P3[i]
if ch = = 3.
exit the loop.

step 7: clean up
- delete P1, P2, P3.

step 8: END.

POLYTECHNIC / DEGREE

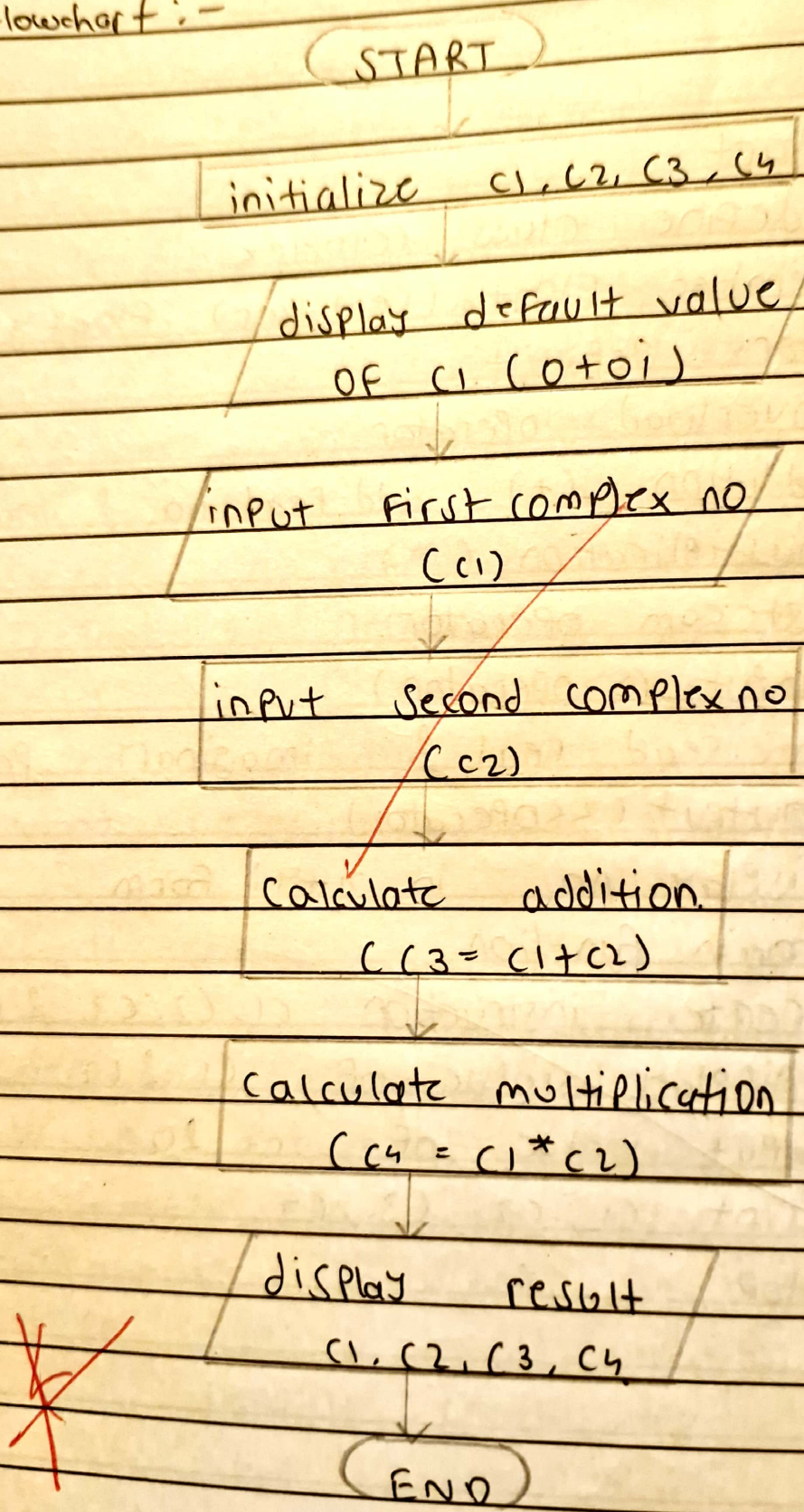
Page :
Date :



Algorithm :

1) Start
2) define classes
   • Publication
   - Attributes : title (String), price (float)
   - constructor : initialize title & price
3) • books (inherits from Publication)
   - Attributes Pages (int)
   • initialize title, price & Pages
   • get b() to get details is bet 500-1500
   - if valid display back details.
4) • Tape
   - min (float) initialize title price & min
   get t() to get details
   if min is not bet 30-90 & price 500-1000.
   - if valid display ~~be~~ details.
5) main function.
   - Ask if they want buy a book (1) or tape (2)
   - if 1, call g.get() for bank.
   if 2, call t.get() for tape.
   catch exception.
6) END.

Flowchart :-



Algorithm:

step 1 : Start
Step 2 : define class complex
- data member float x (real pair), float y.
- constructor : default.
step 3 : overload operator
- Addition (+), add real no & imaginary.
- multiplication (*)
step 4 : stream operator.
- input (>> operator)
for read real & imaginary part.
- output (<< operator)
display no in x+yi form.
step 5 : main function.
- create instruction c1, c2, c3, & c4.
- display value of c1 & c2
- input value of c1 & c2
- Print c1, c2, c3, c4
Step 6 - stop.