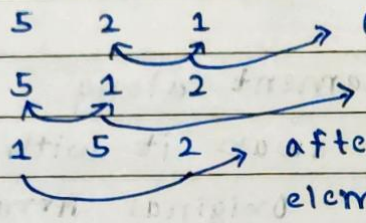
→ THEORY:-

BUBBLE SORT:-

⇒ Bubble sort is one of the simplest and the most popular sorting method. The basic idea behind bubble sort is as a bubble rises up in water, the smallest element goes to the beginning. This method is based on successive selecting the smallest element through exchange of adjacent element.



Out of sequence, hence exchange.

Out of sequence, hence exchange.

after two exchanges, the smallest element has come to the beginning.

Let, n be the number of elements in an array a[]. The first pass begins with the comparsion of a[n-1] and a[n-2]. If a[n-2] is larger than a[n-1], the two elements are exchanged. The smaller elements, now at a[n-2] is compared with a[n-3] and if necessary the elements are exchanged to place the smaller one in a[n-3]

o Pseudo code for Bubble Sort:

Loop for sorting an array a[] having n elements.

```
for (i=1; i<n; i++) {
    for (j=0; j<n; j++) {
        if (a[i] > a[j+1]) {
            exchange a[j] and a[j+1]
```

Outer loop is for passes and the inner loop is for comparsions.

In pass 1 (i=1), there will be n-1 comparisons (j=0 to n-2)

In pass 2 (i=2), there will be n-2 comparisons (j=0 to n-3)

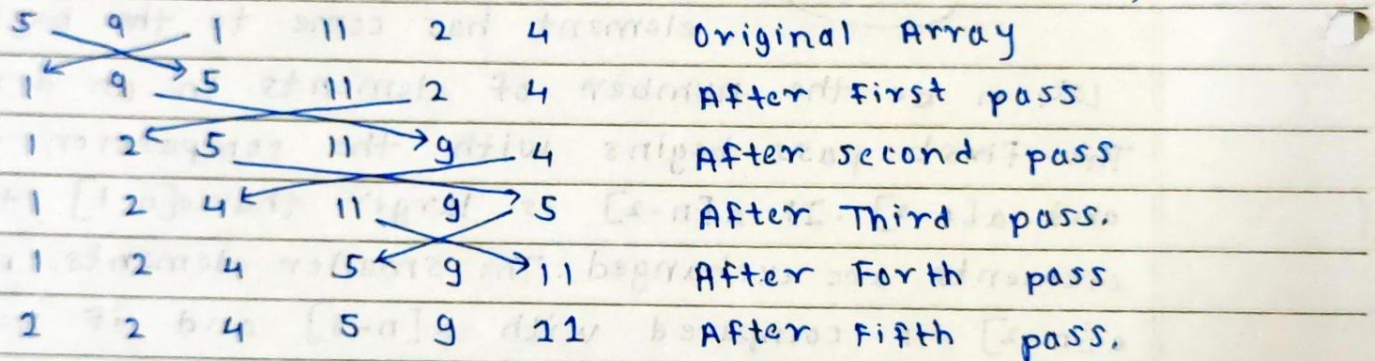
⋮

In pass n-1 (i = n-1), there will be 1 comparison (j=0 to 0)

SELECTION SORT:-

⟹ Selection sort is a very simple sorting method. In the ith pass, we select the element with the lowest value among a[i], a[i+1] ....., a[n-1] and we swap it with a[i]. As a result, after i passes (pass number 0 to i-1) first i elements will be in sorted order. selection sort can be described by:

```
for (i=0 ; i<n ; i++)
    select the smallest element along
        a[i], ....., a[n-1] and swap it with a[i];
```



⟶
```
void selectionSort (int a[], int n)
{  int i, j, k, temp;
   for (i=0 ; i<n-1 ; i++) {
      k=i;
      for (j=i+1 ; j<n ; j++) {
         if (a[j] < a[k]) }
               k=j;
         if (k!=i)
         {
            temp = a[i];
            a[i] = a[k];
            a[k] = temp;
         }
      }
}
```

(1.) Pseudocode for Function for Selection sort of Elements.

→ STEP: 1: start.

STEP: 2: Define a Function def selection_sort(marks):

STEP: 3: Input: List of Marks (marks)

4. for i in range(len(marks)):
(begin) min_idx = i (end)

5. for j in range(i+1, Len(marks)):
(begin) if (marks[min_idx] > marks[j]) then
min_idx = j (end)

6. marks[i], marks[min_idx] = marks[min_idx], marks[i]

7. print The marks of students using Selection Sort.

8. for i in range(len(marks)):
print(marks[i])

9. Stop.

(2.) Pseudocode for Function for Bubble Sort of Elements.

→ 1. Define a Function def Bubble_Sort(marks):

2. Input: n = len(marks)

3. for i in range(n-1):
for j in range(0, n-i-1):
if marks[j] > marks[j+1] then
marks[j], marks[j+1] = marks[j+1], marks[j]

4. print The Marks of students using Bubble Sort

5. for i in range(len(marks)):
print(marks[i])

6. Stop.

(3.) Pseudocode of the Function for displaying top five marks or percentage of students.

→ 1. Start.

2. Define a Function def top_five_marks(marks):

3. print The Top five Marks/percentage.

4. print(* marks[::-1], sep = "\n")
5. Stop.

(4.) Pseudocode for the Main Function.

→ 1. Input 'n' number of students whose marks are to be displayed.
2. Create a list : marks = [ ]
3. for i in range (0, n):
   Input 'ele' i.e. the marks of student
4. marks. append (ele)
5. Stop.

(5.) Pseudocode for Menu / Option List.

→ 1. Intidlize flag = 1.
2. while flag = = 1 :
   print ("\n ---------- MENU ----------")
   print(" 1. Selection sort of the Marks ")
   print (" 2. Bubble Sort of the Marks")
   print (" 3. Exit ")
3. Input: ch = Enter your choice (1 - 3)
4. if ch = = 1 then
   selection_sort(marks)
   a = input ("\n Do you want to display top marks from the list (yes / no) :")
   if ~~then~~ a = = 'yes' then
   top_five_marks (marks)
   else
   print Thanks for using the Program.
   flag = 0
5. elif ch = = 2 then
   Bubble_Sort (marks)
   a = input ("\n Do you want to display top Five marks

4. print(* marks[::-1], sep = "\n")
5. Stop.

(4.) Pseudocode for the Main function.

→ Define main function ⇒ def main():
1. Input 'n' number of students whose marks are to be displayed.
2. Create a list : marks = [ ]
3. for i in range (0, n):
   Input 'ele' i.e. the marks of student
4. marks.append(ele)
5. Stop.

(Continued)

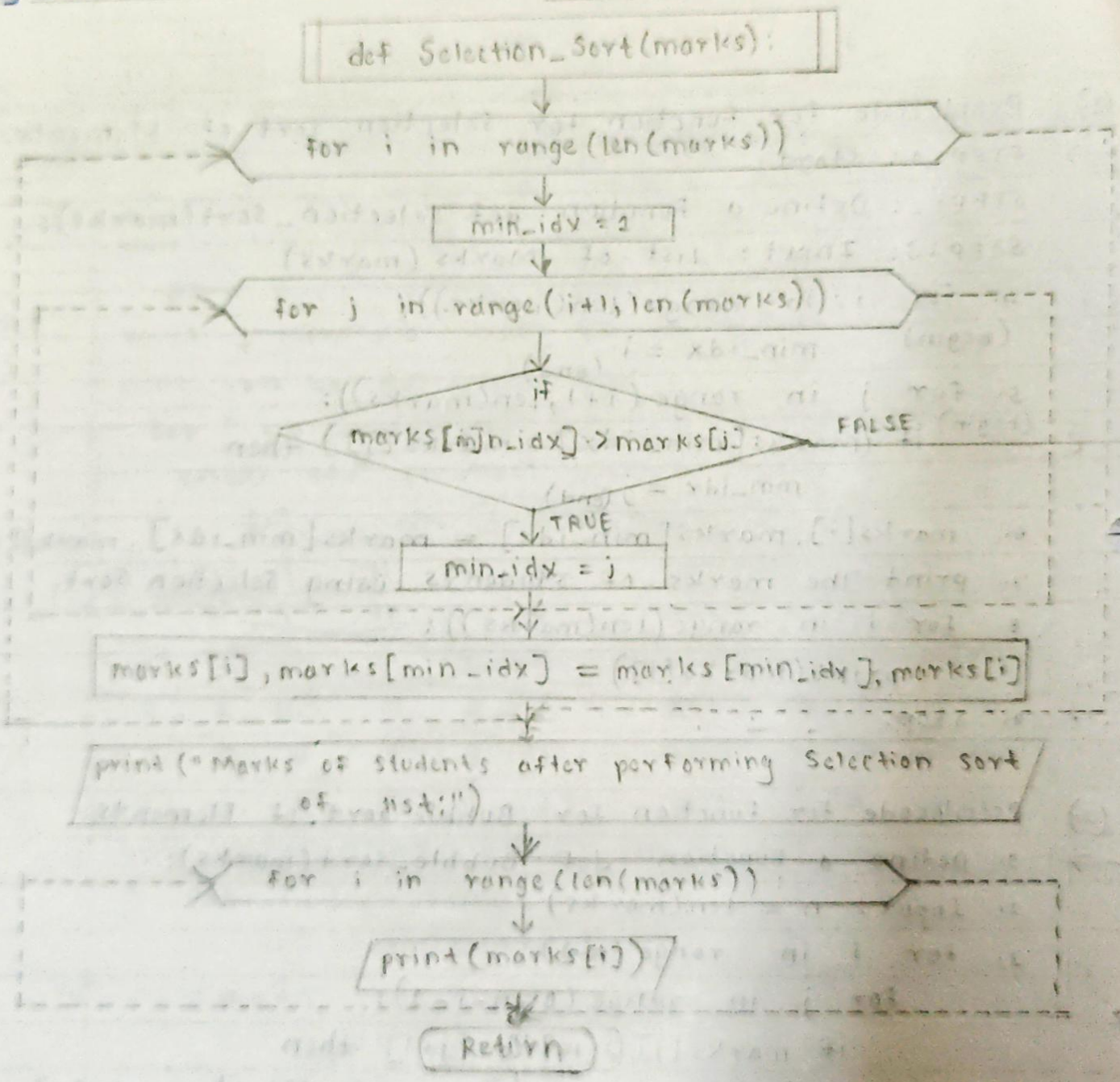
(5.) Pseudocode for Menu/Option List.

→ 1. Intidlize flag = 1.
2. while flag = = 1 :
   print ("\n ---------- MENU ----------")
   print("1. Selection sort of the Marks")
   print("2. Bubble Sort of the Marks")
   print("3. Exit")
3. Input: ch = Enter your choice (1-3)
4. if ch = = 1 then
   selection_sort(marks)
   a = input("\n Do you want to display top marks from the list (yes/no):")
   if ~~then~~ a = = 'yes' then
   top_five_marks(marks)
   else
   print Thanks for using the Program.
   flag = 0
5. elif ch = = 2 then
   Bubble_Sort(marks)
   a = input("\n Do you want to display top five marks from the list (yes/NO):")

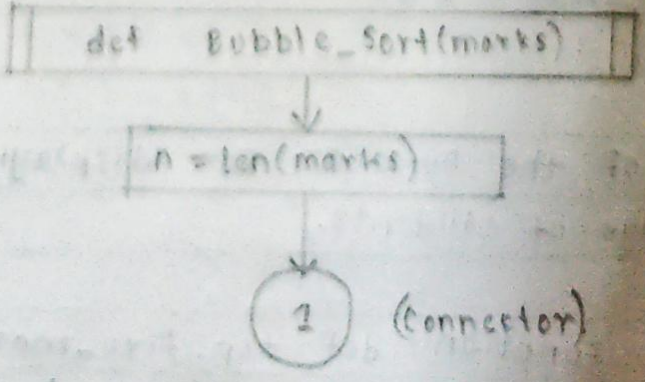
Page No.
Date

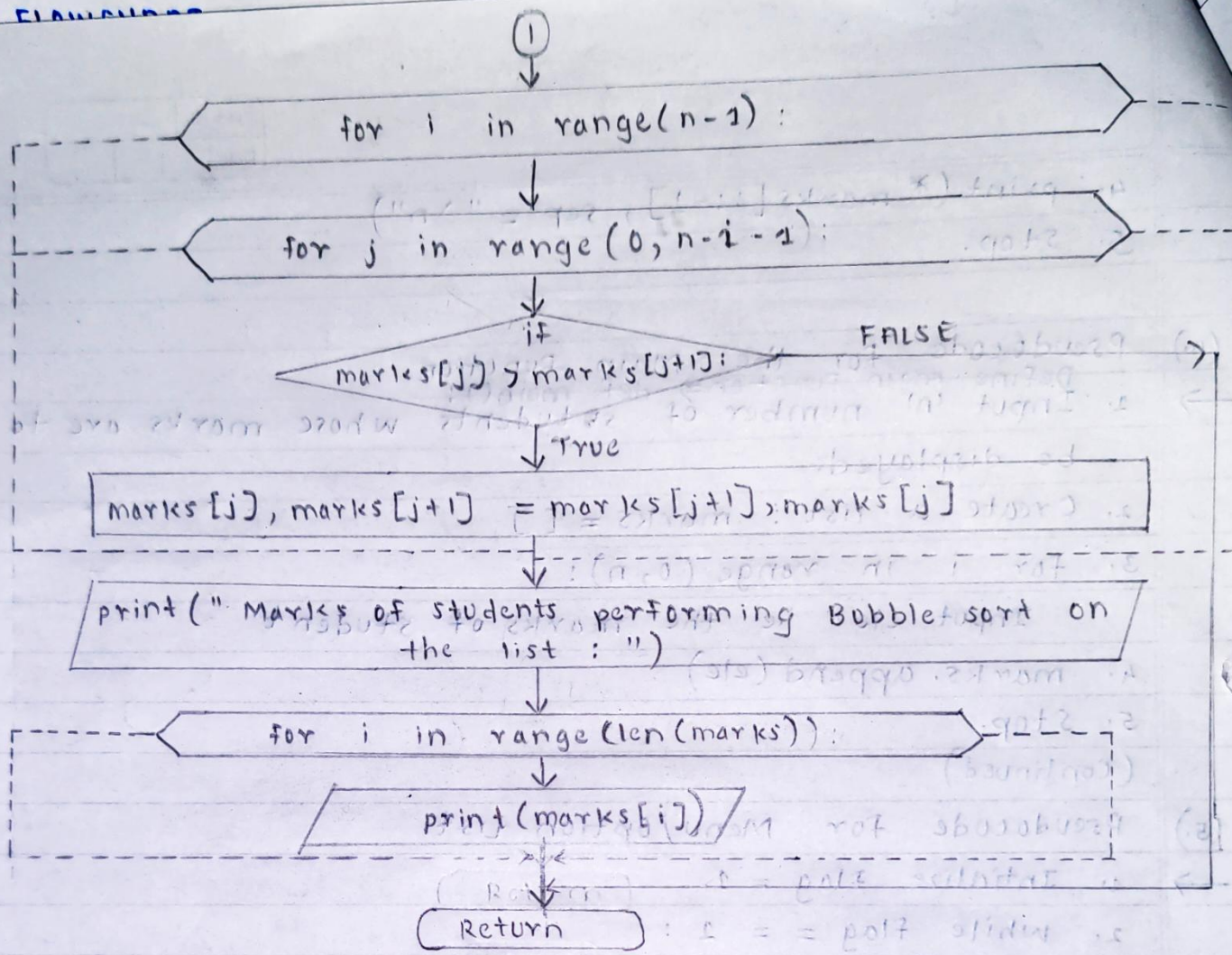
5. if a == 'yes' then
   top_five_marks(marks)
   else
   print Thanks for using the program.
   Flag = 0.
6. elif ch == 3 then
   print Thanks for using the program.
   Flag = 0
7. else
   print Enter a valid choice.
   print Thanks for using the Program.
   flag = 0.
8. main()
9. Stop.

2.) FLOWCHART FOR FUNCTION:- def selection_sort(marks):

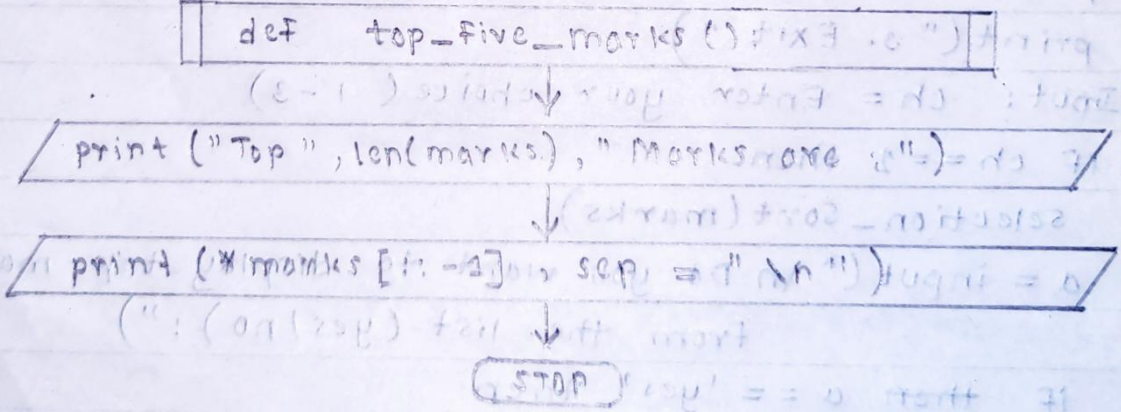


3.) FLOWCHART FOR FUNCTION :- def Bubble_Sort(marks):

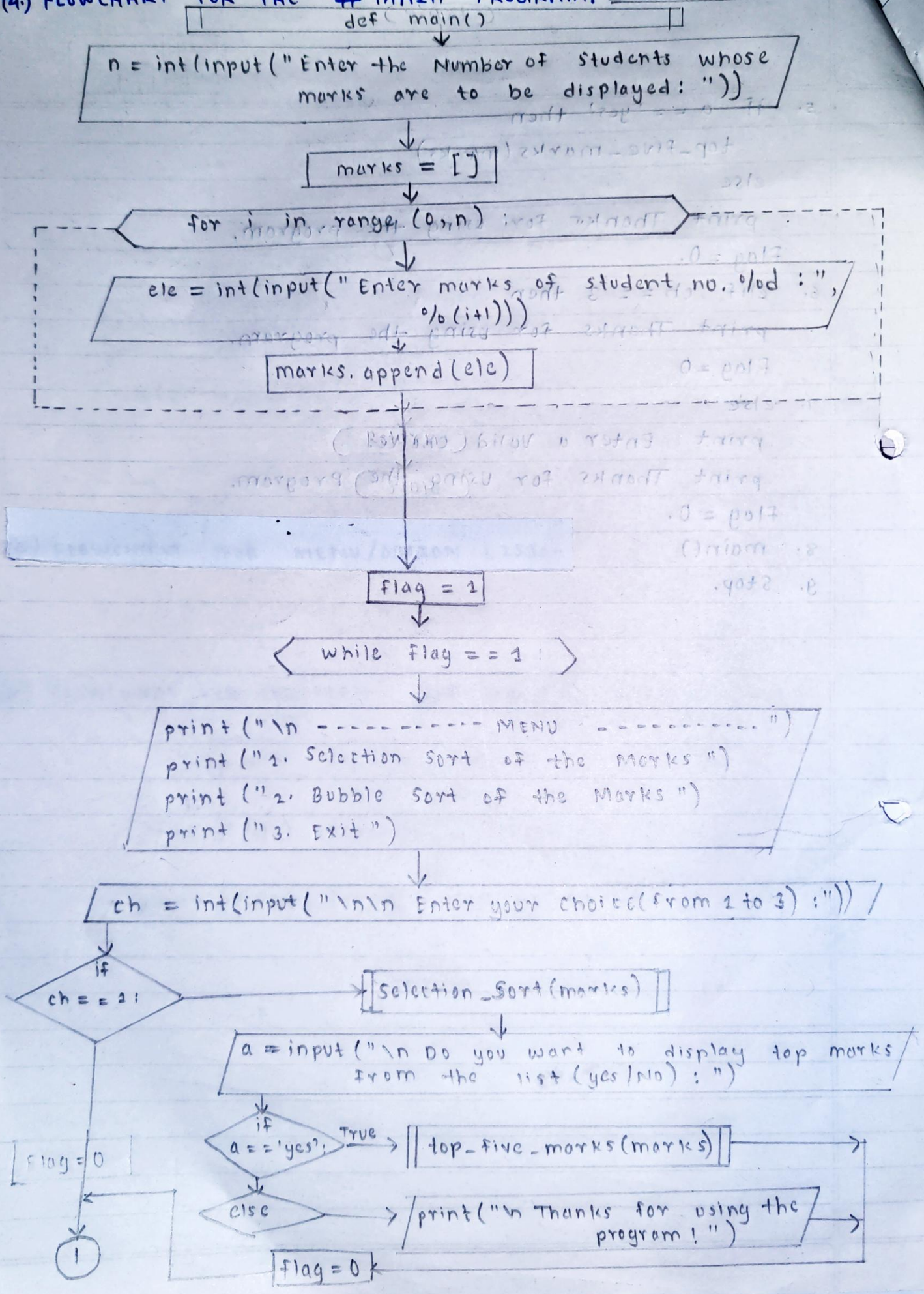


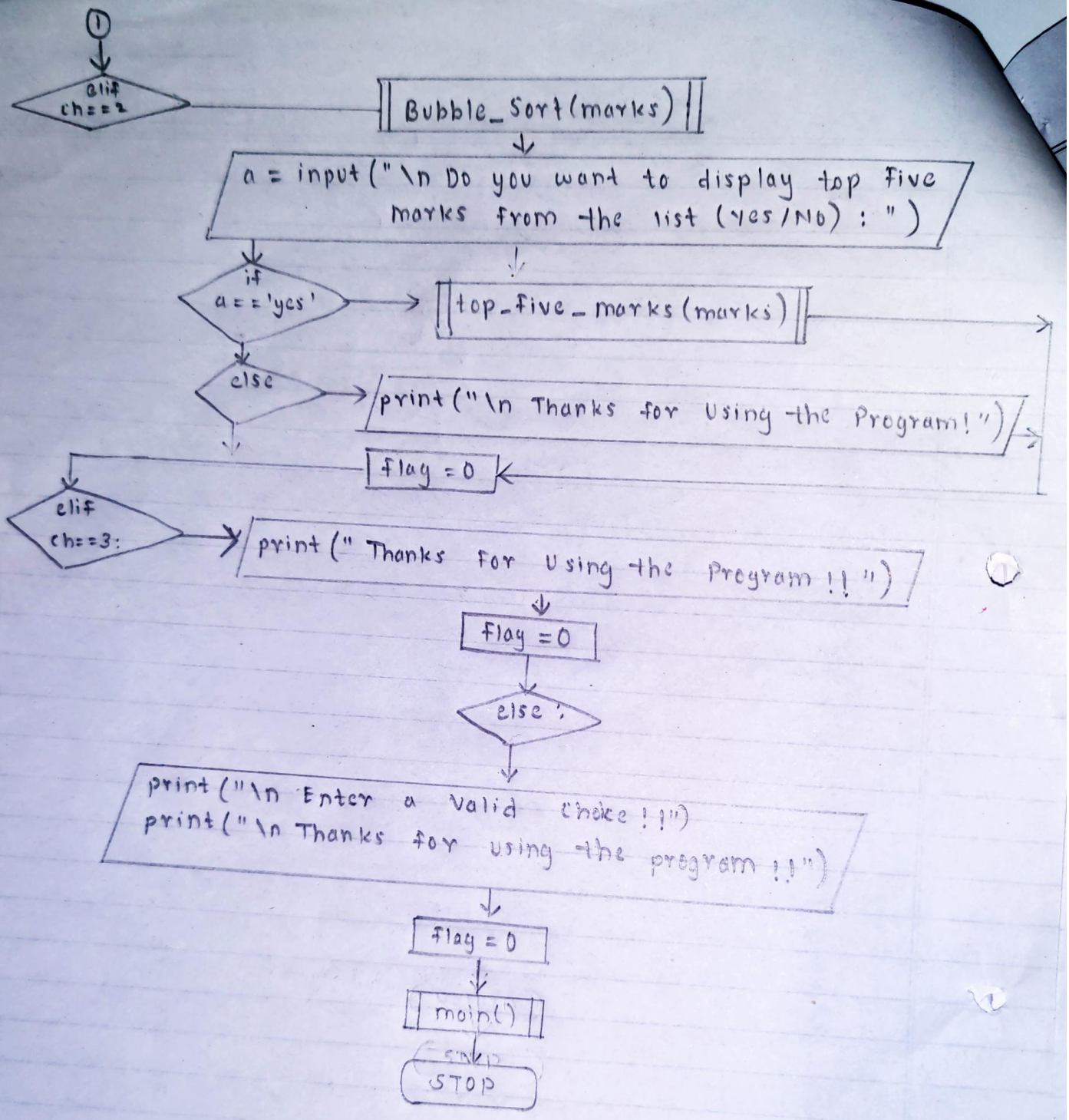


(3.) FLOWCHART FOR FUNCTION: def top_five_marks():



(4.) FLOWCHART FOR THE # MAIN PROGRAM. def main():




1
elif ch==2
Bubble_Sort(marks)
a = input("\n Do you want to display top five marks from the list (yes/NO) : ")
if a=='yes'
top_five_marks(marks)
else
print("\n Thanks for Using the Program!")
flag = 0
elif ch==3:
print(" Thanks for Using the Program !! ")
flag = 0
else :
print("\n Enter a valid choice !!")
print("\n Thanks for using the program !!")
flag = 0
main()
STOP

QUESTIONS:-

Q1] Explain Merge sort with an Example and write C++ program for same.

ANS. Merge sort is a Divide and Conquer Algorithm. It divides the input array into two halves, calls itself for the two halves and then it merges the two sorted halves. The merge() function is used for merging two halves.

The merge(arr, l, m, r) is a key process that assumes that arr[l..m] and arr[m+1, r] are sorted and merges the two sorted sub-arrays into one.

For Example:-

```
#include <iostream>
using namespace std;

// Merge two subarrays L and M int arr
void merge(int arr[], int p, int q, int r) {

  // Create L ← A[p..q] and M ← A[q+1..r]
  int n1 = q - p + 1;
  int n2 = r - q;

  int L[n1], M[n2];

  for (int i=0; i<n1; i++)
    L[i] = arr[p+i];
  for (int j=0; j<n2; j++)
    M[j] = arr[q+1+j];

  int i, j, k;
  i=0; j=0; k=p;
```

```
while (i < n1 && j < n2) {
    if (L[i] <= M[j]) {
        arr[k] = L[i];
        i++;
    } else {
        arr[k] = M[j];
        j++;
    }
    k++;
}

while (i < n1) {
    arr[k] = L[i];
    i++; k++;
}
while (j < n2) {
    arr[k] = M[j];
    j++; k++;
}
}

void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        int m = l + (r-l) / 2;

        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}
```

```
void printArray(int arr[], int size) {
    for (int i=0 ; i<size ; i++)
        cout << arr[i] << " ";
    cout << endl;
}
int main()
{
    int arr[] = { 6, 5, 10, 12, 9, 2 } ;
    int size = sizeof(arr) / sizeof(arr[0]);

    mergeSort(arr, 0, size - 1);

    cout << " Sorted Array : \n";
    printArray(arr, size);
    return 0;
}
```