

Name: Natasha Saldanne.

USN: 4AUGCS057.

Implement merge Sort.

Algorithm:-

1) Start.

2) Input $a[] = \{64, 82, 45, 105, 125, 268, 15, 268, 1, 2, 25, 28\}$

3) merge (int **a, int i begin, int i mid, int i end, int **b)

4) splitmerge (int **b, int i begin, int i end, int **c)

5) **merge sort (int *a, int size)

6) int size = size of a / size of a [0]

7) int **ret = merge sort (a, size)

8) Display sorted array elements

9) For (int i = 0; i < size; i++)

Display *ret [i]

10) Free (ret)

11) stop.

merge (int **a, int i begin, int i mid, int i end, int **b)

1) Entry

2) int i = i begin, j = i mid

3) For (int k = i begin; k < i end; k++)

$b[k] = a[i < mid \ \&\& \ (j \geq i end \ || \ *a[i] \leq *a[j])$
 $i++ ; j++$

4) End.

Split merge (int **b, int i begin, int i end, int **c)

1) Entry

2) if (i end - i begin < 2)

return.

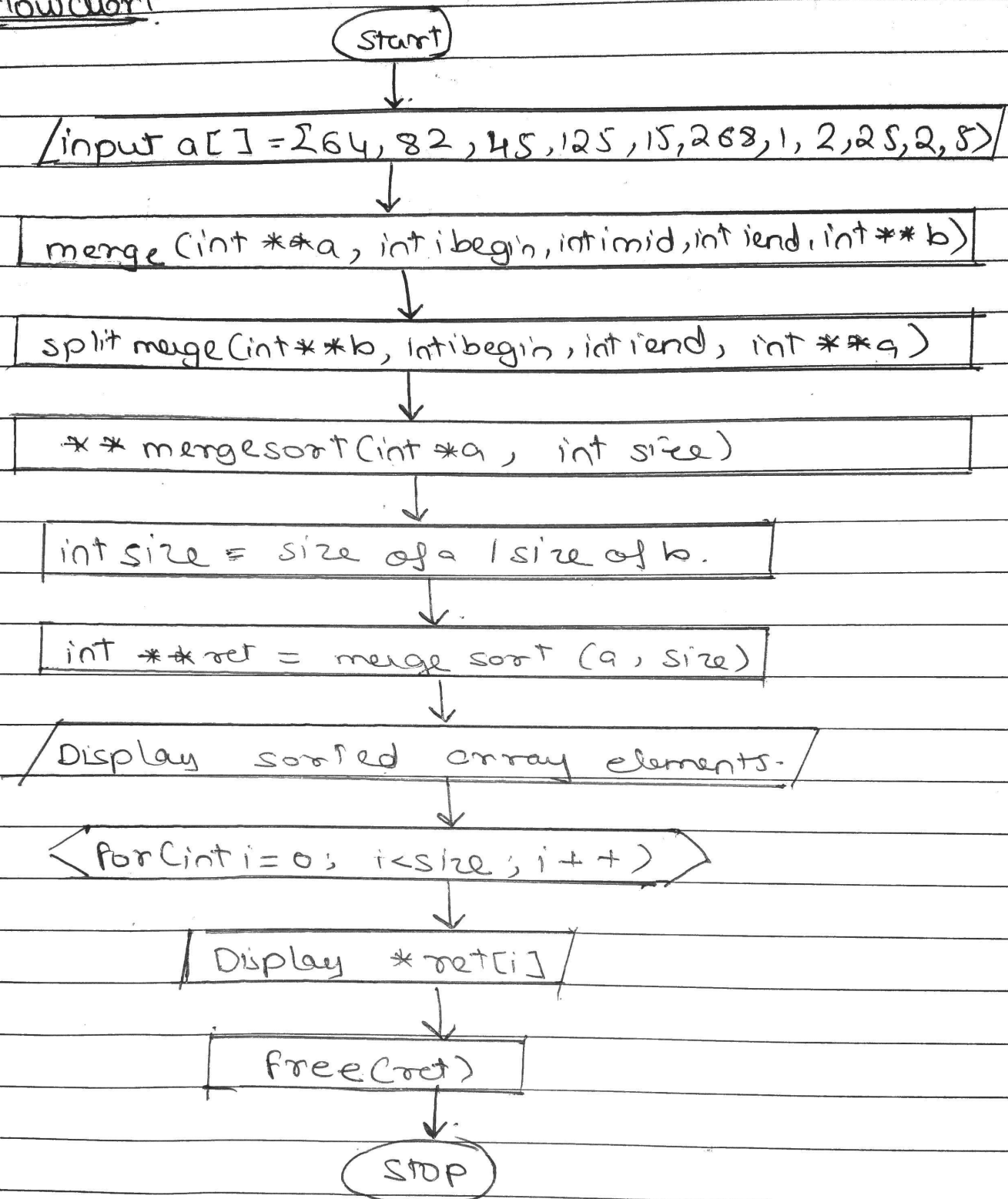
3) int i mid = (i end + i begin) / 2.

- 4) split merge (a, ibegin, imid, b)
- 5) split merge (a, imid, iend, b)
- 6) merge (b, ibegin, imid, iend, a)
- 7) End.

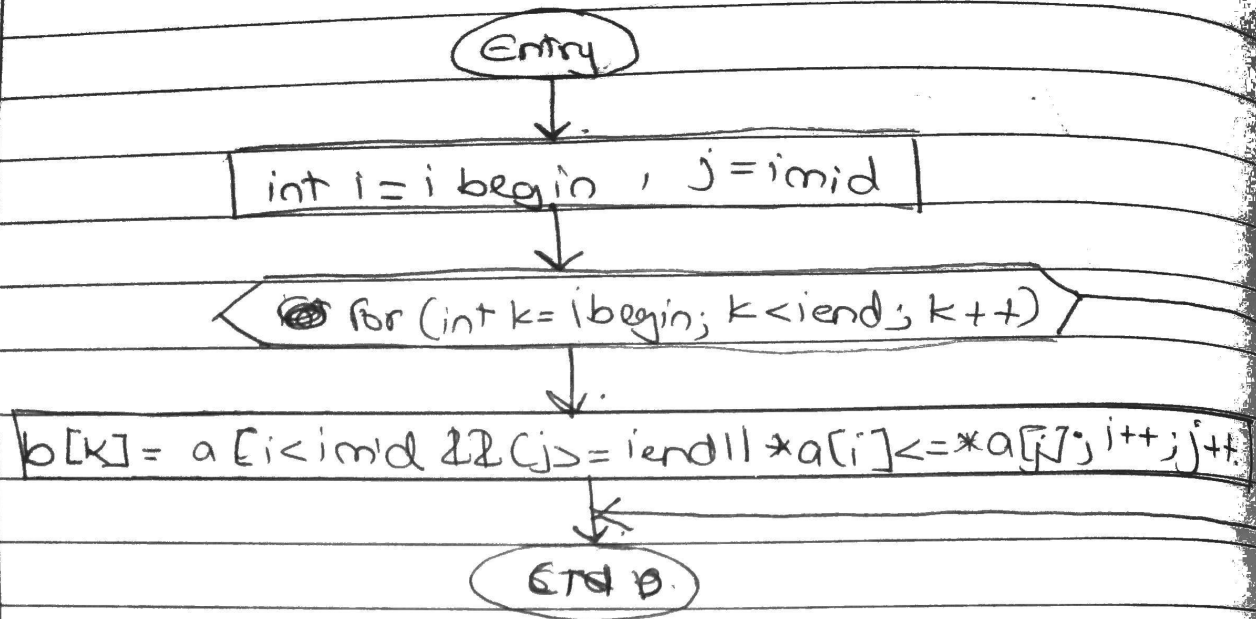
** merge sort (int * a, int size)

- 1) Entry.
- 2) int ** ret = malloc (size * sizeof * ret)
- 3) int ** temp = malloc (size * sizeof * temp)
- 4) for (int i=0; i < size; i++)
ret[i] = temp[i] = a+i.
- 5) split merge (temp, 0, size, ret)
- 6) free (temp)
- 7) return ret

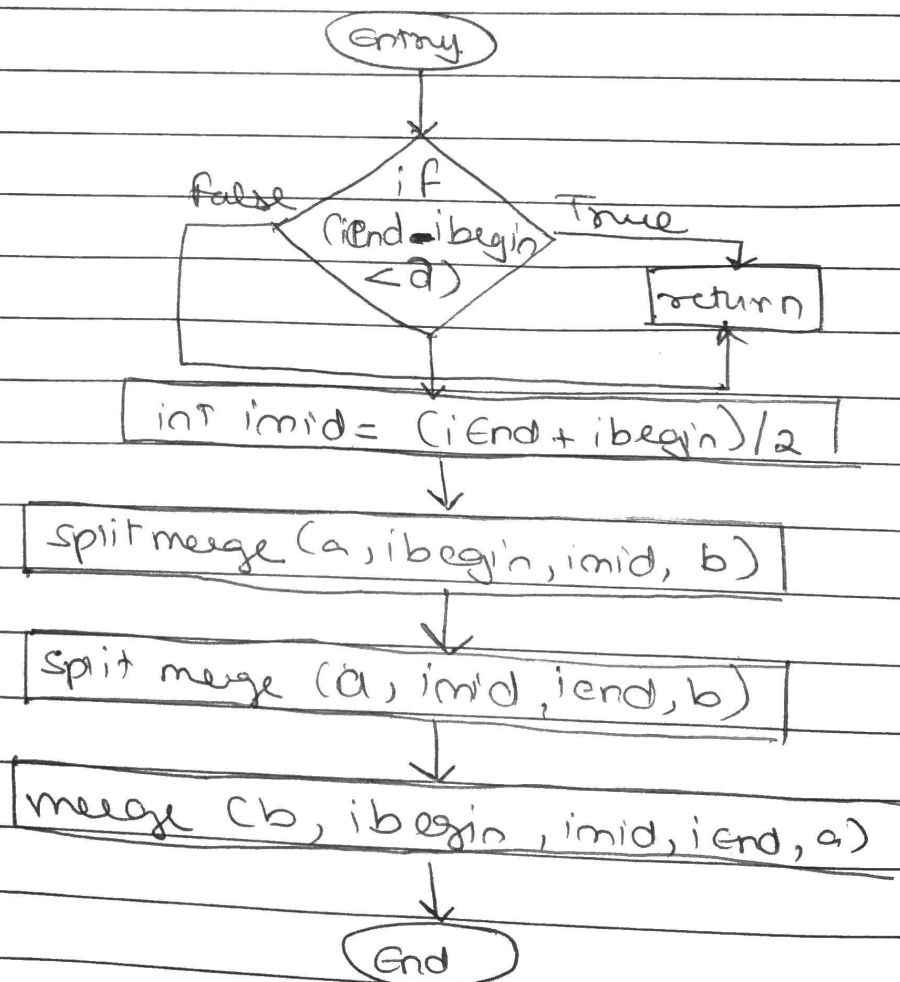
Flowchart



merge (int **a, int i begin, int i mid, int i end, int **b)



split merge (int **a, int i begin, int i end, int **b)



**merge sort (int *a, int size)

