GAYATHRI

4AL19CS035

Algorithm :-

Step1: Start

Step2: Input a[] = {64, 52, 45, 125, 15, 268, 1, 2, 25, 2, 5}

Step3: Merge (int **a, int begin, int mid, int end, int **b)

Step4: SplitMerge (int **b, int begin, int end, int **a)

Step5: **mergesort (int *a, int size)

Step6: int size = size of a / size of a[0]

Step7: int **ret = merge sort (a, size)

Step8: Display sorted array elements

Step9: for (int i=0; i<size; i++)

- Display *ret[i]

Step10: Free (ret)

Step11: Stop.

Step5: Stop

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

Step1: Entry
Step2: int *i = *begin, j = *mid
Step3: for (int k = *begin; k < *end; k++)

b[k] = a[*i < mid && (j >= *end || **a[*i] <= **a[j]) ? *i++ : j++]

Step4: End

Split merge (int **b, int *begin, int *end, int **a)

Step1: Entry
Step2: if (*end - *begin < 2)

return

Step3: int *mid = (*end + *begin)/2
Step4: Split merge (a, *begin, *mid, b)
Step5: Split merge (a, *mid, *end, b)
Step6: merge (b, *begin, *mid, *end, a)
Step7: End

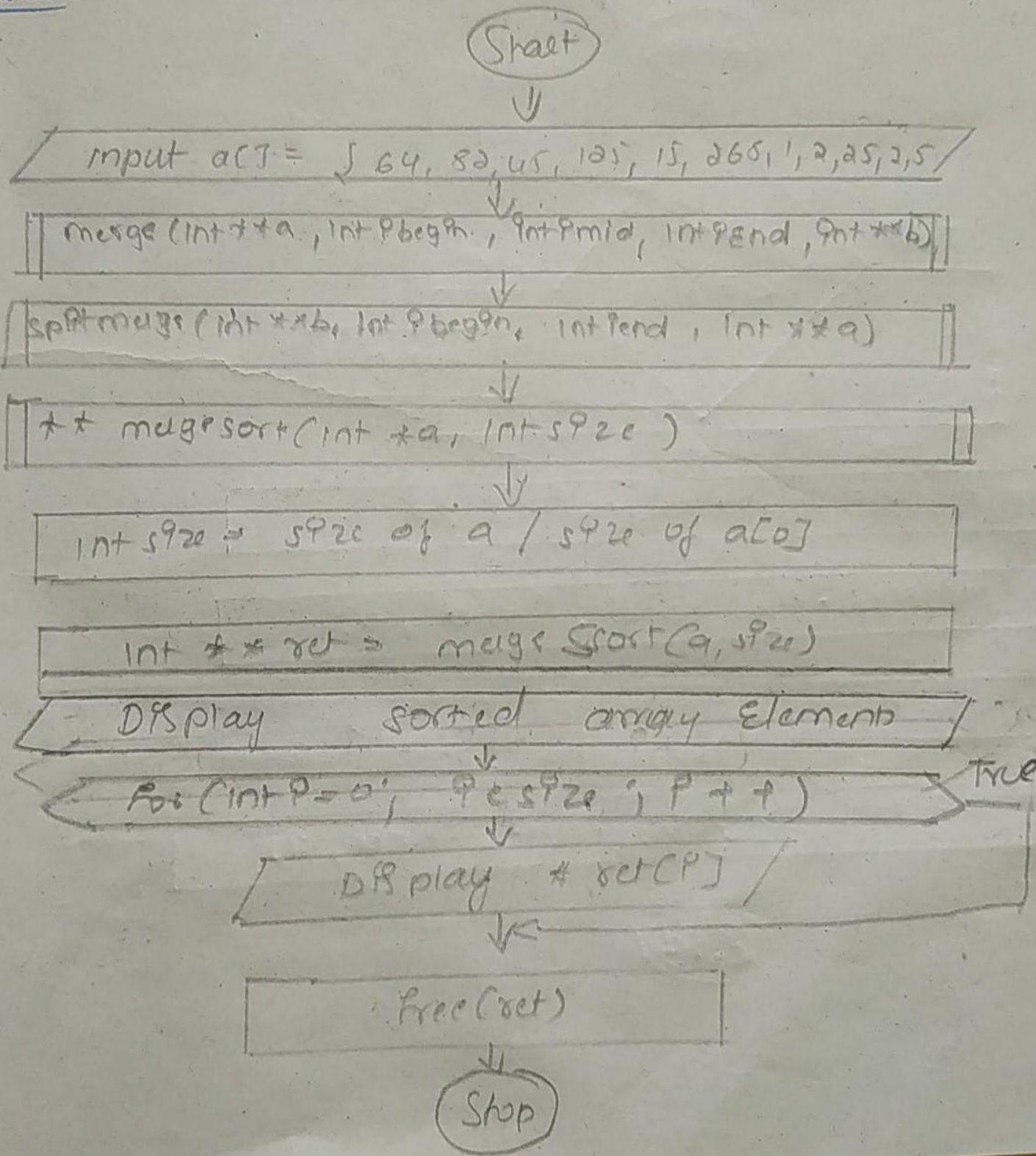
** merge sort (int *a, int size)

Step1: Entry
Step2: int **ret = (int **) malloc (size * sizeof *ret)
Step3: int **temp = malloc (size * sizeof *temp)
Step4: for (int i = 0; i < size; i++)

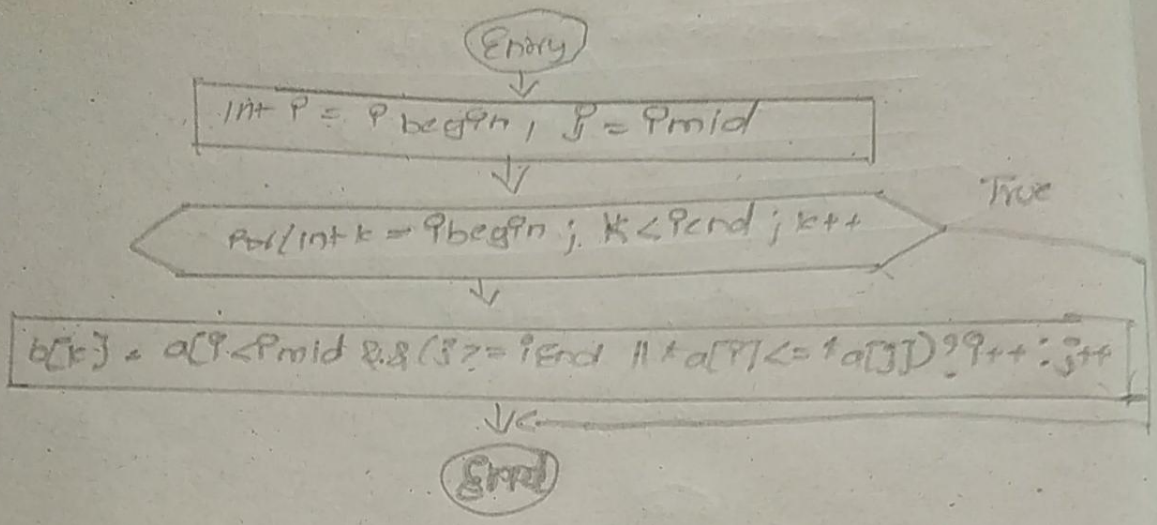
ret[i] = temp[i] = a+i

Step5: Split merge (temp, 0, size, ret)
Step6: free (tmp)
Step7: return ret

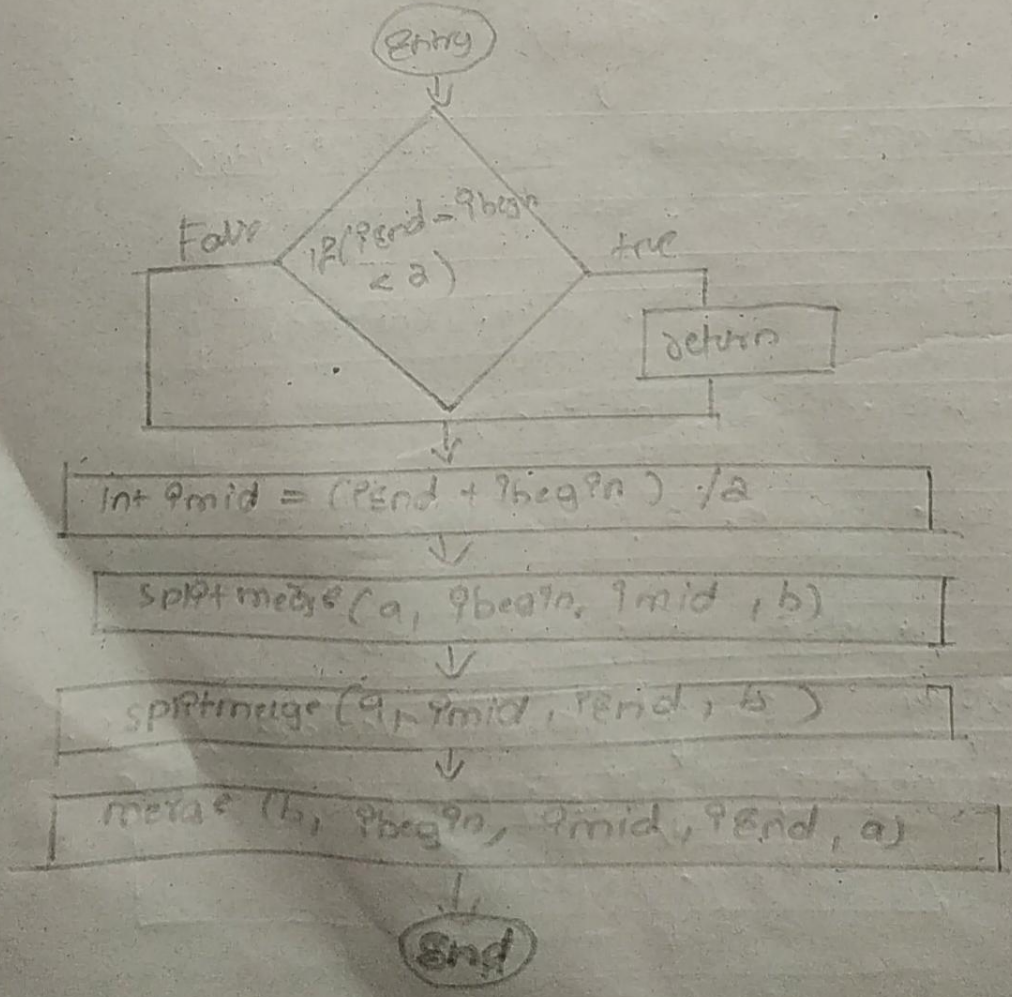
Flowchart


Start
input a[] = {64, 82, 45, 101, 15, 265, 1, 2, 25, 2, 5}
merge(int **a, int *begin, int *mid, int *end, int **b)
splitmerge(int **b, int *begin, int *end, int **a)
** mergesort(int *a, int size)
int size = size of a / size of a[0]
int ** ret = mergeSort(a, size)
Display sorted array Elements
For(int i=0; i<size; i++)
True
Display *ret[i]
Free(ret)
Stop

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



splitmerge(int **b, int *begin, int *end, int ***a)



** mergeSort(int *a, int size)