MAIN()

```
While 1
    Scan c          //Menu
    Scan n          //No of students
    Switch c
        Case r : Call read_students(A, n)
                 Break
        Case p:  Call print_students(A, n)
                 Break
        Case a:  Call arrange_students(A, n, pos)
                 Break
        Case d:  Call divide_students(A, 0, n, rval)
                Break
        Case l:   scan rval
                 list_students(A, n, rval)
                 Break
        Case t:  return 0
```

READ_STUDENTS(A,n)

```
For i=1 to A.length
    Read A[i]
```

PRINT_STUDENTS(A, n)

```
For i=1 to A.length
    Print A[i]
```

ARRANGE_STUDENTS(A, n,pos)

```
k=0
For i=1 to pos and  j=pos+1 to A.length
    B[k]=A[i]
    B[k+1]=A[j]
    k+=2
For i=0 to A.length
   A[i]=B[i]
```

DIVIDE_STUDENTS(A, l, n, rval)

```
char p;
    p=partition(A,low,high);
    divide_students(A,low,p);
    divide_students(A,p+1,high);
```

PARTITION(A, l, h,rval)

```
 Pivot =rval
i= l
j=h
While i<j
   Do
      i=i+1
   While A[i]<=pivot
   Do
      j=j+1
   While A[i]>pivot
If i<j
    temp=A[i]
    A[i]=A[j]
    A[j]=temp
temp=A[l]
A[l]=A[j]
A[j]=temp
Return j
```

LIST_STUDENTS(A,n,rval)

```
count=0
For i=1 to A.length
   If A[i]==rval
       count=count+1
      print i
      A[i]=999000 // JUNK OR SENTINEL VALUE
 If count==0
print -1
```