Insertion sort

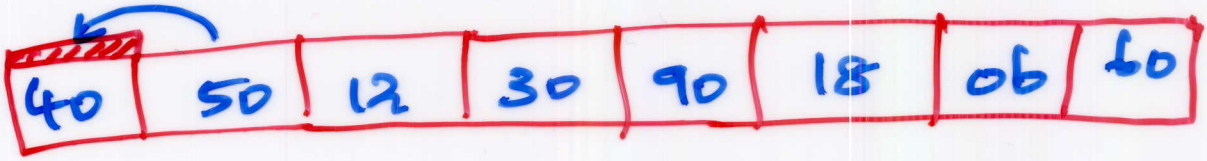
- inserts each item in the proper place in the final list.
- an element (first element) in the unsorted part is inserted in the appropriate place of the sorted part
- I iteration: element at posn 2 is compared with the element at posn 1.
- II iteration: element at posn 3 is compared with the elements in 2nd and 1st posns.
- Repeated upto n-1 iterations

Ex:

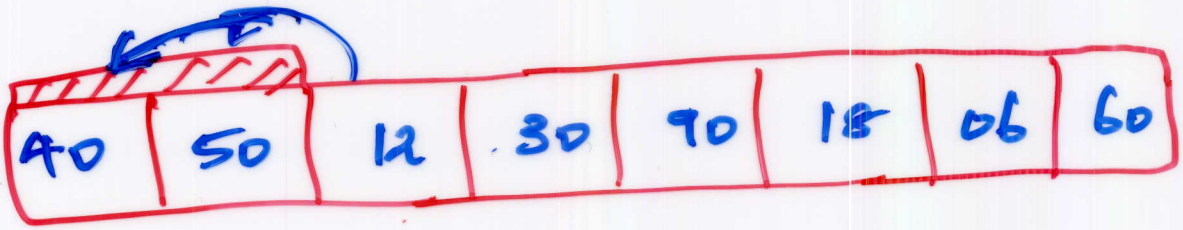
40 50 12 30 90 18 06 60

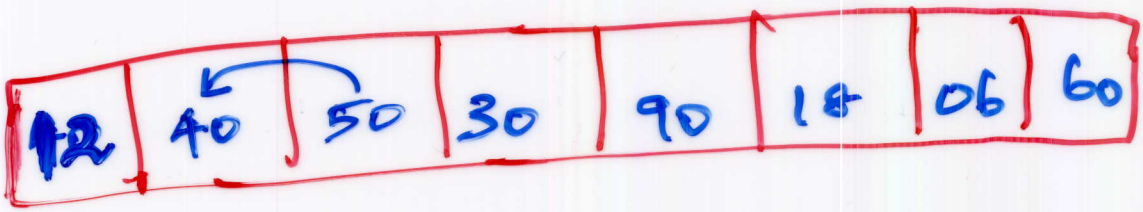
needs 7 iterations

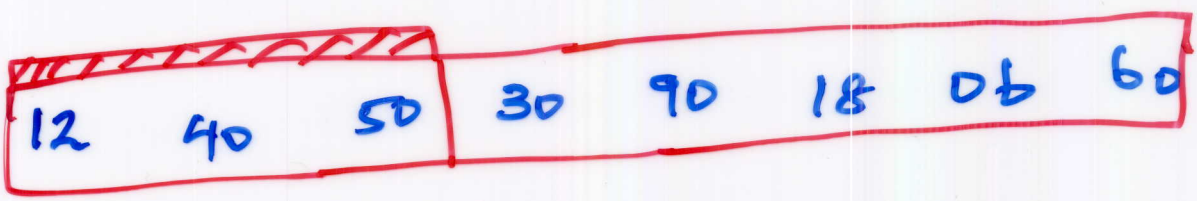
I



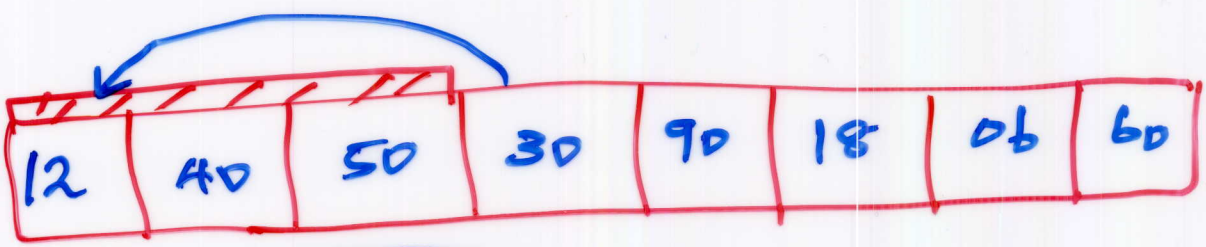
II

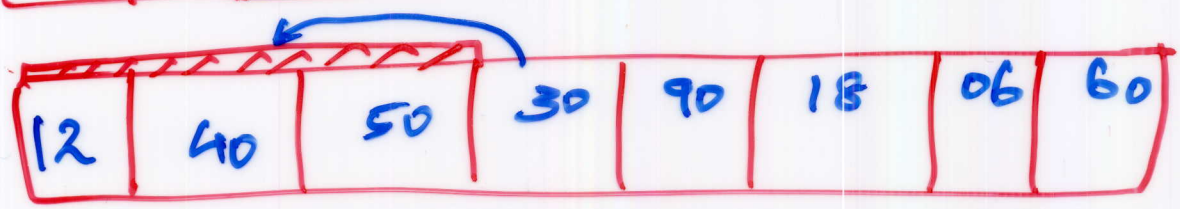


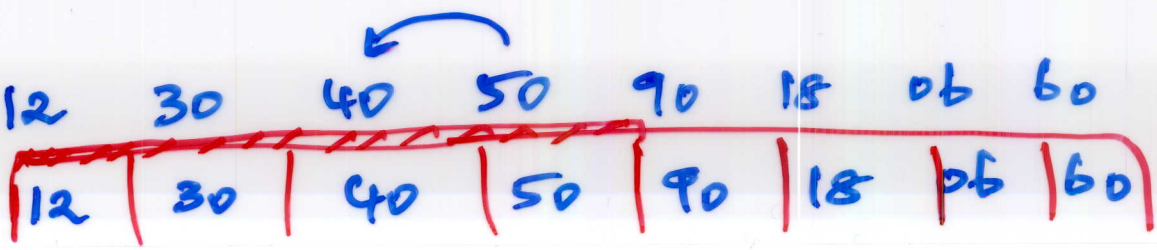




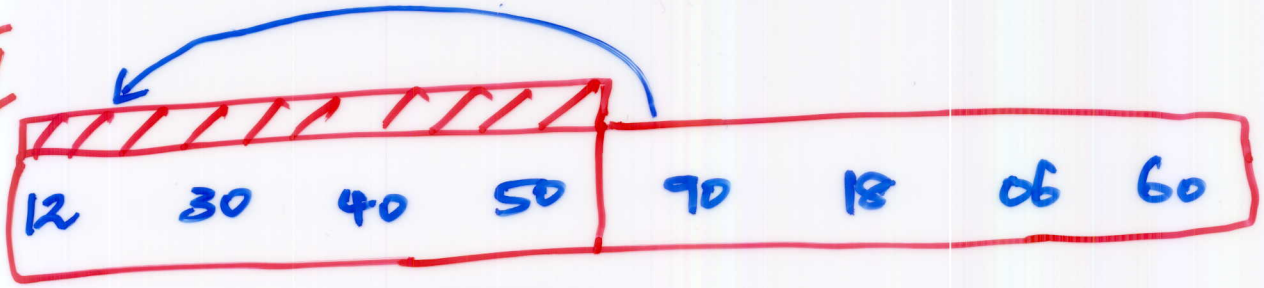
III

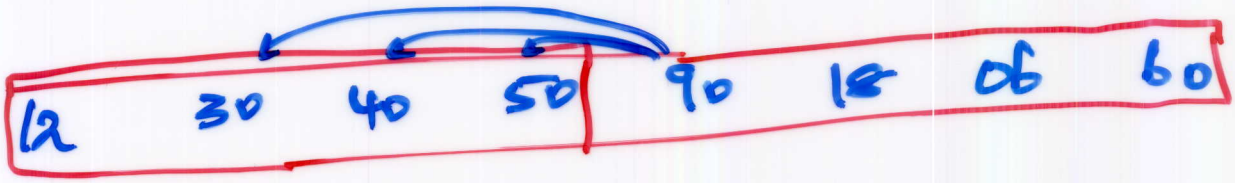






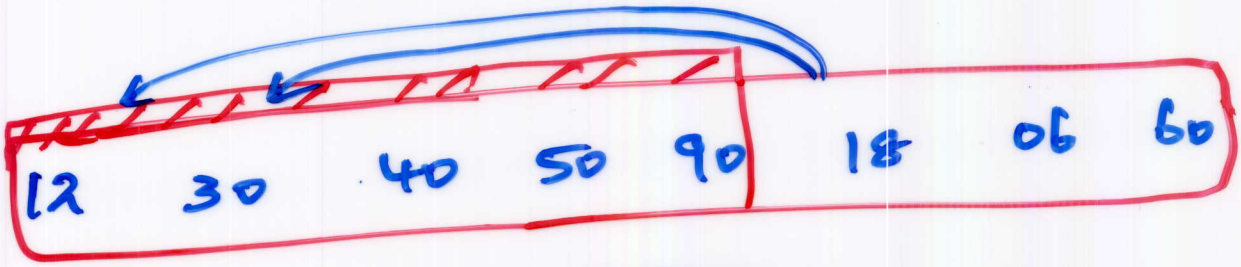
IV


12 30 40 50 90 18 06 60

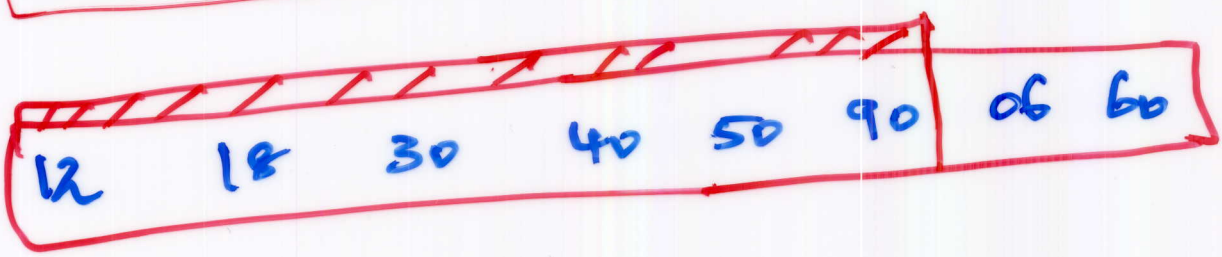

12 30 40 50 90 18 06 60


12 30 40 50 90 18 06 60

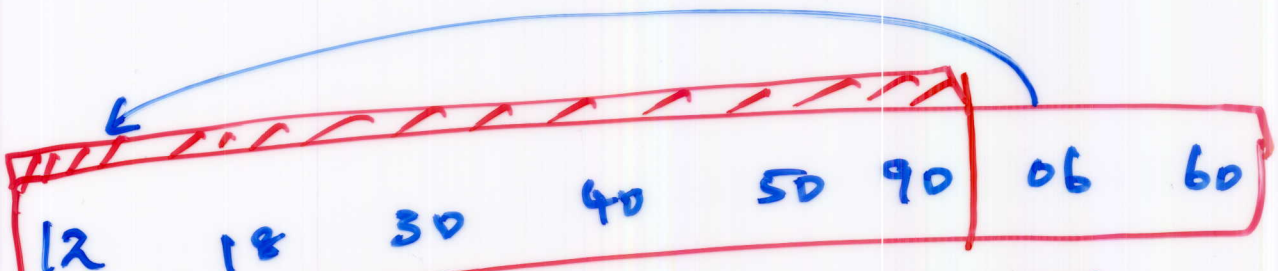
V

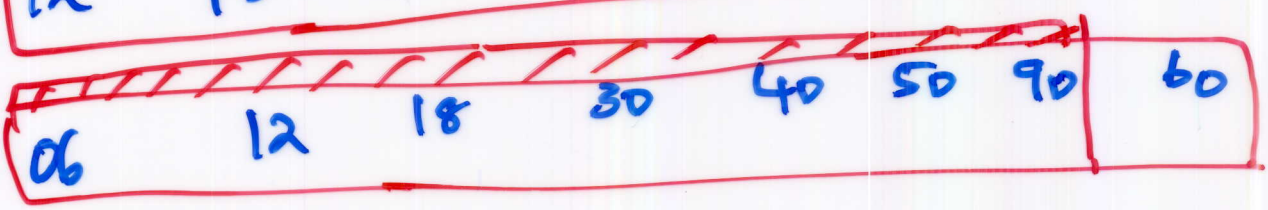

12 30 40 50 90 18 06 60


12 18 30 40 50 90 06 60

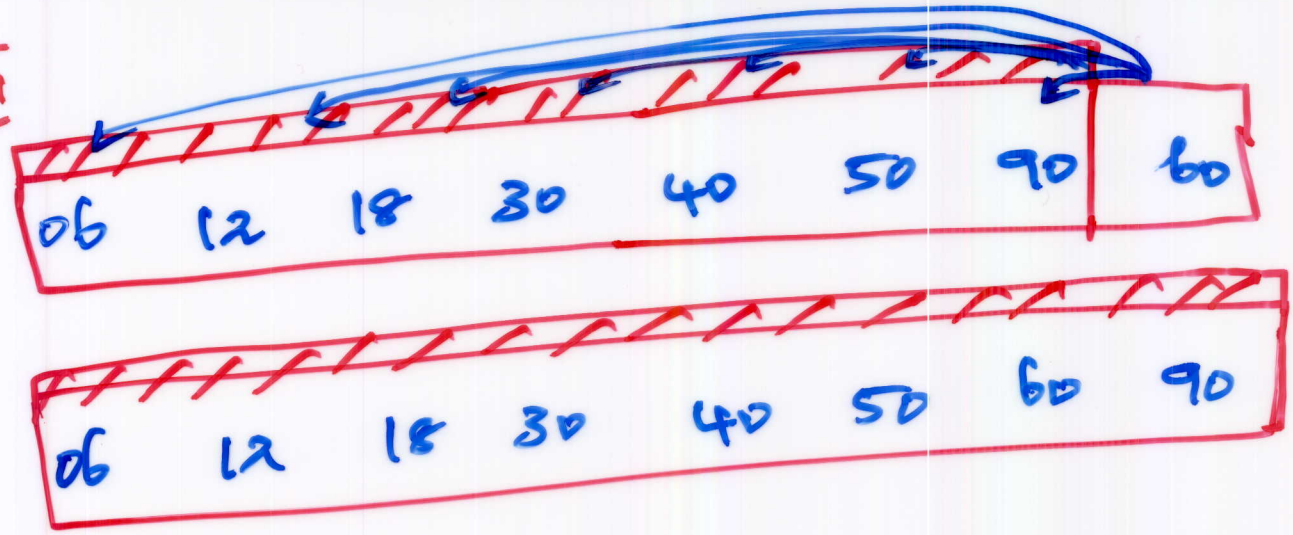

12 18 30 40 50 90 06 60

VI


12 18 30 40 50 90 06 60


06 12 18 30 40 50 90 60

VII



Insertion-sort(A)

```
1.  for j ← 2 to length(A)
2.      do key ← A[j]
3.      /* insert A[j] into the sorted *
4.      /* sequence A[1..j-1] */
5.          i ← j-1
6.          while i>0 and A[i]>key
7.              do A[i+1] ← A[i]
                   i ← i-1
8.
9.          A[i+1] ← key
```

write a program to sort a list of N numbers using insertion sort algo. The necessary data must be read from a sequential file.

```
#include <stdio.h>
#include <conio.h>
main()
{
  int n=0, i, k, m=0;
  int a[100]
  char fname[10];
  FILE *fp;
  clrscr();
  printf("Enter File name");
  scanf("%s", fname);
  fp = fopen(fname, "r");
  if (fp == NULL)
```

```
{
  printf("Can't open file\n");
  exit(0);
}
printf("The numbers in file are");
while (!feof(fp))
{
  fscanf(fp, "%d", &a[n]);
  printf("%d\n", a[n]);
  n++;
}
for (i=0; i<n; i++)
{
  temp = a[i];
  for (j=i-1; j>=0; j--)
  {
    if (temp < a[j])
      a[j+1] = a[j];
    else
      break;
  }
  a[j+1] = temp;
}
```

```
printf("Sorted array...\n");
for(i=0; i<n; i++)
{
    printf("%d\n", a[i]);
}
}
```

NOTE:

Create the file and store the list to be sorted in it, before running this pgm.