Assignment

G. Karthik
AP19110010329
CSE-F

1.

1(a)

```
#include<stdio.h>
Void binary-search();
int a[50];n items,toe,beg,mid,end,i;
Void main()
{
 Print f("\nenter the size of an array");
 Sconf("%d",&n);
Printf("\n enter element of array in shorted
            form:\n");
 for (i=0;i<n;i++)
  Sconf("%d",&a[i]);
 Printf("\n enter item to be Searched:");
 Sconf("%d",&item);
 binary-Search();
 getch();
 {
   void binary-Search();
 {  beg=0
    end=n-1
   mid=(beg+end)/2;
while ((beg<=end) && (a[mid]!=item))
{
  if (item<a[mid])
     end=mid-1
  else
     beg=mid+1
     mid=(beg+end)/2
}
 if (a[mid]==item)
 Printf ("\n\n Item Found at location %d",mid+1)
                                             ;
else
 Printf ("\n\n item doesn't exist");
 }
```

```
(b) #include <stdio.h>
      int main()
      {
        int arr[10];
          int sum, Product, i;
     Printf("\n enter elements :\n");
       for (i=0; i<10; i++)
```

```
{
 Printf ("enter arr [%d]:",i);
  Scanf ("%d", & arr [i]);
}
  Sum=0;
    Product=1;
    for (i=0; i<0; i++)
  {
    Sum= Sum+arr[i];
    Product = Product * arr [i];
  }
  Printf ("\n Sum of array is %d", sum)
   Printf ("\n Sum of array is %d", Produc
  return 0;
  }
```

2.
```
# include<stdio.h>
# include<stdio.h>

 // merge two subarray of arr[]
 // First subarray is arr [~~m+~~l, ---r]
 // Second subarray is arr [m+1, --r]
```

```
void merge (int arr[], int l, int m, int r)
{
     int i, j, k;
     int n1 = m-l+1;
     int n2 = r-m;
     Int 2 [n1], R [n2];
     For (i=0; i<n1; i++)
       2(i) = arr [l+i]
      for (j=0; i<n2; j++)
         R[i] = arr [m+1+j];
   l=0; (intial index of 1st subarray)
   J=0; (intial index of end subarray)
   K=1; (initial index of merge subarray)
  while (i<n1 && j<n2);
  {
       if (L[i] <= R[j])
   {
      ~~else~~ arr[k] = L[i];
          i++;
    {
      else
    {  arr [K] = R[j];
```

```
j++;
}
k++;
}
while (j<n2)
{
  arr[k]=R[j];
   j++;
   k++;
}
Void mergesort(int arr[],int l,int r)
{
    if (l<r)
   {
     int m=l+(r-l)/2;
    merge sort (arr, l, m);
    merge sort (arr, m+1, r);
    merge (arr, l, m, r);
   }
 }
void Print array (int A[], int size
{
int i;
For (i=0; i<size; i++)
```

```
Printf ("%.d", r[i]);
 printf ("\n");
{
 int main ()
 {
 int arr [ ] = [1.2, 11, 13, 5, 6, 7};
 int arr-size = size of (arr)\ size of arr[0]);
   Printf (" Given array is \n")
   ~~Print (" Given arrao~~
    Print array (arr, 0, arr-size-1);
   Print (" \n sorted array is\n");
   Print array (arr, arr-size);
  return 0;
 }
```

3. Selection sort :

```
 #include <stdio.h>
   void swap (int*a, int* b)
  {   int temp = *a;
      *a = *b
      *b = temp
```

```
}
void selection sort (int array [], int size
 {
   for (int step=0; step<size-1; step++)
   {
     int (min-idx=step;
    For (int i=step+1; i<size, i++)
   {
     if (array [i] < array imin-idx])
     min-idx=i;
    }
    swap( & array [min-idx], & array (step]);
    }
   }
   void Print array (int array [], int size)
    {
      for (int i=0; i< size, i++) {
      Printf ("%d", array [i]);
    }
    Printf ("\n");
     }
    }
```

```
int main()
 {
 int data[] = {20,12,10,15,2};
  int size=size of data /size (data[0]);
   Selection sort (data, size)
  Printf ("sorted array in asscending order
                                   :\n");
   Print array (data,sizes);
  }
```

```
3. # include <math.h>
   # include <stdio.h>
   void insertion sort (int arr[],int n)
    {
      int i,.Key j;
       For (i=1; i<n; i++)
      {
        Key=arr[i]
         j=i-1
       while (j>=0 && arr[i]>Key)
      {
        arr[j+1] = arr[i];
```

```
  j = j - 1
 }
   arr [j+1] Key;
  }
 }
void Print array (int arr[], intn)
 {
  int i;
  for (i=0; i<n; i++)
    Print f ("%d", arr[i]);
    Print f ("\n");
  }
    int main()
    {
     int arr[] = {12, 11, 13, 5, 6);
    Int n=size of(arr) [size of arr(0]];
     Insertion sort (arr, n);
     Print array (arr, n);
      returno;
     }
```

```
4(i) #include<stdio.h>
     # include <math.h>
     int main()
     {
     int a[], {16,19,11,15,10,12,14};
     int i,j;
      For(j=0;i<7;j++)
      {  int swapped=0;
       i=0;
       while(i<7-1)
       { if (a[i]>a[i+1])
           int temp=a[i];
            a[i]=a[i+1];
            a[i+1]=temp;
           swapped=1;
         { i++;
        } if(swapped)
       break;
      }
}
 for (i=0; i<7; i++)
   Point f ("%d\n", a[i]);
   return 0;
  }
```

```
4(ii) # include <stdio.h>
      # include <math.h>
      Int main()
      {
       int a[7] = {16,9,11,15,10,12,14
       Int i,j;
         For (j=0; j<7; j++)
       {
          int swapped = 0;
          i = 0;
          while (i<7-1)
          {
             if (a[i] > a[i+1])
          {   int temp = a[i];
              a[i] = a[i+j];
a[i+1] =temp;
   swapped=1;
  }i++;
  }
   if (!swapped)
   break;
   }  for(i=0; i<7; i++)
      printf("%d\n",a[i]);
     return 0;
     }
```

```
4(iii) #include <stdio.h>
       #include <conio.h>
     {
       int num,even_sum=0,odd prod=1,
  run,temp;
   printf("enter any number:");
       scanf("%d", &num);
   while (mm>0)
   { num= num %10;
```

```
 if(rem % 2==0)
  even sum =even sum+rem;
else
 odd Prod = odd Prod* rem;
  num=num/10;
}
Printf("\n Sum of even digit= %d",even
Print("\n Product of odd digit=%d"odd
  getch();
   return0;
  }
4(iii) # include <stdio.h)
  void swap(int* xP,int* yP)
  {
    int temP=*xP;
    *xP=*yP;
    *uP=temP;
  }
```

```
int i,j;
 for (i=0; i<n-1; i++)
  for (j=0; j<n-i-1; i++)
  if (arr[j]>arr[j+1])
   Swap(& arr[j] & arr[j+1]);

 }
  void Print array (int arr[], int size)
 {
  int i;
  for (i=0; i<size, i++)
  Printf("%d", arr[i]);
   Printf["\n"];
  }
   int main ()
  {
   int arr[]= {64,34,25,12,22,11,90};
  int n= sizeoff (arr)/size of arr(0));
   bubble sort (arr,n);
    Print f("sorted array:\n");
  Print array (arr;n),
   return 0;
  }
```

```
5. #include <stdio.h>
   void binary_search(int [],int,int,int);
   void bubble_sort(int[],int);
   int main()
  {
    int key, size, i;
    int list [25];
   Printf("enter size of a list");
    Scanf("%d",& size);
    Printf("enter elements\n");
    for(i=0; i<size, i++)
    {
      Scanf ["%d",& size s+[i])
    }
      bubble_sort(list,size);
     Printf ("\n");
      Printf("enter key to search\n")
      Scanf("%d", key);
    binary search (list,0, size key);
```

```
} void bubble . search (list, 0, size, key);
{
    int temp. i,j;.
   {
     ~~Void~~ 1
     for (i=0; 1<size; i++)
     {
       for (j=i, j<size ; j++)
       {
           if (list (1) > list [j])
         {
           temp= list [i]; aq
          list [i] = list [j]
            list [j] = temp;

     {
   {
  {
 {
void binary search (int list [], int lo, int hi, intkey
 {
   int maid ;
     if (lo > hi)
   { printf ("key not found \n");
} else if (list [mid] > key)
{
    binary.search (list, lo, mid -1, key);
 }
   clese if (midj < key)
  }
   binary - search (list, mid + i, hi, ke
 }
}
```