DSA - Assignment 6

M. Mohan Sukumar
AP19110010467

1) a)

```
#include <stdio.h>
void binary_search();
int a[50], n, item, loc, beg, mid, end, i;
void main()
{
 printf("\n Enter the size of an array");
 scanf("%d", &n);
printf("\n Enter elements of an array in
     sorted form: \n");
for(i=0; i<n; i++)
   scanf("%d", &a[i]);
printf("\n Entel ITEM to be searched:");
scanf("%d", &item);
 binary_search();
 getch();
}
void binary_search();
{
  beg=0
  end=n-1
  mid = (loc+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 doesnt 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< 10; i++)
{
sum = sum + arr[i];
Product = Product * arr[i];
```

```
printf("\n Sum of array is : %d", sum);
printf("\n product of array is : %d", \n product);
Return 0;
}
```

② 
```
#include <stdio.h>
#include <stdio.h>
void merge (int arr[], int l, int m, int r)
{
  int i, j, k;
  int n1 = m - l + 1;
  int n2 = r - m;
  int L[n1], R[n2];
  for (i=0; i<n1; i++)
  L[i] = arr[l+i];
  for (j=0; j<n2; j++)
    R[i] = arr[m+1+j];
i = 0;   (initial index of 1st subarray)
j = 0;   (initial index of 2nd subarray)
k = l;
while (i<n1 and k<r j<n2)
{
  if (L[i] <= R[j])
    { arr[k] = L[i];
      i++;
```

```
    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;
   mergesort (arr, l, m);
   mergesort (arr, m+1, r);
   merge (arr, l, m, r);
  }
}
void printarray (int A(), int size()
{int i;
 for(i=0; i<size; i++)
  Printf("%d", A[i]);
 Printf("\n");
```

```
{
    int arr[] = {12,11,13,5,6,7};
    int arr_size = sizeof(arr)/sizeof(arr[0]);
    printf("Given array is \n");
    printArray(arr, 0, arr_size-1);
    print("\n sorted array is \n");
    printarray(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[min_idx])
        min_idx = i;
    }
    swap(&array[min_idx], &array[step]);
  }
}
void printarray(int array[], int size)
{
  for(int i=0; i<size; i++)
  {
    printf("%d", array[i]);
  }
  printf("\n");
}
}
int main()
{
  int data[] = {20,12,10,15,1};
  int size = sizeof(data) / sizeof(data[0]);
  selectionsort(data, size)
  printf("sorted array in ascending order\n");
  printarray(data, size);
}
```

3) #include <math.h>

#include <stdio.h>

void insertionsort( int arr[], int n)

{ int i, key, j;

for (i=1; i<n; i++)

{

key = arr[i];

j = i-1

while (j >= 0 && arr[j] > key)

{

arr[j+1] = arr[j];

j = j-1

}

arr[j+1], key;

}

}

void printarray (int arr[], int n)

{ int i;

for (i=0 ; i<n; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main()

{ int arr[] = {12,11,15,5,6};

```
int n= sizeof (arr)/sizeof arr[0];
insertionsort(arr,n);
printarray (arr, n);
return 0;
}
```

④ 1) `#include <stdio.h>`

```
#include <math.h>
int main()
{ int a[] = {16,19,11,15,10,12,14};
 int i,j;
 for(i=0; i<7;i++)
 {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)
```

```
}
for(i=0; i<7 ; i++)
    printf("%d\n", a[i]);
    return 0;
}
Void bubblesort(in )
```

(4) ii)

```
#include <stdio.h>
#include <conio.h>
{int num, evensum=0, oddprod=1, temp;
printf("Enter any number:");
scanf("%d", &num);
while (num>0)
{
    num=num%10;
if(num%10==0)
    evensum = evensum+num;
else
    oddprod = oddprod*num;
    num = num/10;
}
```

```
scanf("%d", &size);
printf("Enter elements\n");
for(i=0; i<size; i++)
{ scanf("%d", &list[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;
for(i=0; i<size; i++)
{ for(j=i; i<size; i++)
{ if( list[i] > list[j])
{
temp = list[i]
```

```
    printf("%d", arr[i]);
    printf("\n");
}
int main()
{ int arr[] = {64, 34, 25, 12, 22, 11, 90};
  int n = sizeof(arr)/sizeof(arr[0]);
  bubbleSort(arr, n);
  printf("Sorted array: \n");
  printArray(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", &list[i])
}
bubble_sort(list, size);
printf("\n");
printf("Entry key to search\n");
scanf("%d", key);
binary seach (list, 0, sizekey);
}
void buble_seach (list=0, size, key);
{int temp, i, j;
for(i=0; i<size; i++)
{for(j=i; i<size; i++)
{if(list[i] > list[j])
{
temp = list[i]
```

```
        lst[i]=lst[j]
        lst[j]=temp;
      }
    }
  }
}
void binary_search (int lst[], int lo, int hi, int key)
{
    int void;
    if (lo > hi).
    {
      print(" key not found");
      defun 0;
    }
mid = ( lo+hi )/2;
if (ist[mid] == key)
{
  printf( "key found \n");
}
elseif( lst[mid] >key]
{ binary_search (lst, lo, mid-1, key);
}
elseif( lst[mid ckey}
```

```
{
    binary_search (lst, midtl, li, key);
}
}
```