Ass-6

Ch. Sushma
AP19110010345
CSE-F

(1a)
```
#include <stdio.h>
#include <conio.h>
void main( ){
clrscr( );
int A[10], n, i, L=0, U=9, f=0, M;
printf ("Enter 10 elements of an array in
        ascending order: \n");
for (i=0; i<10; i++)
scanf ("%d", &A[i]);
printf ("Enter element to be searched
        in an array:");
scanf ("%d", &num);
while (L<U) {
  M = (L+U)/2
if (num > A[M])
  L=M+1;
```

```
else
    if (num > A[M])
        l = M+1;
else
    if (num < A[M])
        u = M-1;
else {
    f=1
    break; }
}
if (f==0)
printf ("%d is not present in array", num);
else
printf ("%d present in array", n, M+1);
getch();
}
```

(b)

```
#include stdio.h
void main () {
int A[10], n, i, l=0, u=9, f=0, M, sum=0, pro
printf ("Enter 10 elements of an array in
        Ascending order \n");
```

```
for (i=0, i<; i++)
scanf ("%d", &A[i]);
printf ("Enter to element be searched:");
scanf ("%d, &n);
while (L <=U) {
    M = (L+U)/2
if (n>A[M])
    L=M+1;
else
  if(n<A[M])
    U= M-1;
else {
    f=1
    break; }
}
while (L<=U){
    M= (L+U)/2
if ( n>A[M])
    L=M+1;
else
  if(n<A[M])
    U= M-1;
else {
    f = 1
    break; }
}
```

(2)

```
#include <stdio.h>
void main() {
int A[5], B[5], C[10];
int i, j, k, temp;
printf ("Enter 5 elements of 1st array");
for (i=0; i<5; i++)
  scanf ("%d", &A[i])
printf ("Enter 5 elements of 2nd array")
  for (i=0; i<5; i++)
  scanf ("%d", &B[i]);
for (i=0; i<10; i++); {
  for (j=i+1; j<5; j++) {
    if (A[i] > A[j]) {
      temp = A[i];
      A[i] = A[j];
      A[j] = temp; }
```

```
if (B[i] > B[j]){
  temp = B[i];
   B[i] = B[j];
    B[j] = temp; } }
for(i=0; j=0; k=0, i<10; i++)
{
 if (A[j] <= B[k]) {
    C[i] = A[j];
       j++;
  }
  else
{
    C[i] = B[k];
       k++; }
if (j==5 || k==5){
    i++;
    break; } }
for (k<5); {
    C[i] = B[k];
    i++;
    k++; }
printf ("Sorted
array using merge
sort.");
for (i=0; i<10; i++)
printf ("%d ", c[i]);
}
```

(3) Insertion Sort :-

Insertion sort is a simple sorting algorithm that builds that fabricates sorted 1 item at an time. This is less efficient on large list than more advanced algorithms such as Quick sort, merge sort etc.

ex:- Input elements : 89 17 8 12 0

Step-1 :- 89 17 8 12 0

[Bold elements are sorted list & non-bold unsorte

step-2 :- 17 89 8 12 0

[Each element will be removed from Unsorted li & placed at right position in sorted list]

step-3 :- 8 17 89 12 0

step-4 :- 8 12 17 89 0

step-5 :- 0 8 12 17 89

Selection sort :-

Selection sort is a simple sorting algorithm. This algorithm is an in-place Comparison based algorithm in which the list is divided into 2 parts, the sorted part at left & Unsorted part at right end.

Ex:- Array : [10, 5, 2, 1]

The 1st element = 10. In next, we must find smallest number for the remaining array. The smallest number = 1, so, we replace 10 by 1

The new array = [1, 5, 2, 10]

Again process repeates

Finally sorted array = [1, 2, 5, 10]

---

(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 list: ");
scanf("%d", &size);
printf("Enter elements ");
for(i=0; i<size; i++){
scanf("%d, &list[i]); }
bubble_sort(list, size);
printf("\n");
printf("Enter key to search");
scanf("%d", &key);
binary - search(list, 0, size, key);
}
void bubble_sort(int list[], int size)
{ int temp, i, j;
for(i=0; i<size; i++){
for(j=0; j<size; j++){
if(list[i] > list[j]){
temp = list[i];
list[i] = list[j];
list[j] = temp; }}}
```

```
}
Void binary_sea
(int list[], int lo
int key){
int mid;
if(lo > hi){
printf("key foun
return;
}
mid = (lo+hi)/
if(list[mid] == key
printf("key found
}
else if(list[mid]<k
{
binary_search(list,
mid-1, key); }
else if
(list[mid] < key){
bina
```

binary_search(list, mid+1, hi, key); }}

④ (i)

```
#include<stdio.h>
void main(){
int arry[100], n, a, b, swap;
printf("Enter no of elments");
scanf("%d", &n);
printf("Enter %d integers", n);
for(a=0; a<n; a++)
 scanf("%d", &array[a]);
for(a=0; a<n-1; a++){
for(b=0; b<n-a-1; b++)
{
 if(array[b] > array[b+1]){
  swap = array[b];
   array[b] = array[b+1];
    array[b+1]=swap;}}
Printf("Sorted list in ascending order:");
      for(a=0; a<n; a++)
printf("%d", array[a]);
printf("Alternate
elements of array:");
for(a=0; a<10; a+=2)
 printf("%d", array[a]);
}
```

(ii)
```
#include <stdio.h>
void main() {
int array[100], n, a, b, swap, sum=0, product=1;
printf("Enter no. of elements:");
scanf("%d", &n);
printf("Enter %d integers\n", n);
for(a=0; a<n; a++)
scanf("%d", &array[a]);
for(a=0; a<n-1; a++) {
for(b=0; b<n-a-1; b++) {
if(array[b] > array[b+1]) {
swap = array[b];
array[b] = array[b+1];
array[b+1] = swap; } } }
printf("Sorted list in ascending order:");
for(a=0; a<n; a++)
printf("%d", array[a]);
printf("Enter upto 5 values:");
for(a=0; a<5; a++)
```

```
scanf("%d", &array[a]);
for(a=0; a<5; a++) {
 if (array[a]%1==0)
        sum=sum+array[a]; }
printf("Total Sum of Odd values is: %d", sum);
 for(a=0; a<5; a++){
  if(array[a]%2==0)
     product = array[a]; }
printf("Total Sum of even values is: %d", product);
}
```

(iii)

```
include <stdio.h>
 void main(){
int array[100], n, a, b, swap, count=0;
 printf("Enter no of elments: ");
 scanf("%d", &n);
 printf("Enter %d intergers: ", n);
 for(a=0; a<n; a++)
 scanf("%d", &array[a]);
 for(a=0, a<n-1; a++){
```

```
for (b=0; b<n-a-1; b++) {
 if (array[b] > array[b+1]) {
    swap = array[b];
     array[b] =  array[b+1];
     array[b+1] = swap; } } }
printf ("Sorted list in ascending order: ");
    for (a=0; a<n; a++)
 Print  printf ("%d", array[a]);
 printf ("Enter the numbers:");
 Scanf ("%d", &m);
 for (a=0; a<n; a++) {
 if (array[a] % m == 0)
      count++; }
  printf ("Total elements divisible by %d
                    %d \n", m, count);
 }
```