DSA Assignment - 6

G. Sumanth Varma
AP19110010208
CSE-G

①
```
#include <stdio.h>
int main()
{
int i, low, high, mid, n, key, arr[100], temp, i, one,
 two, sum, product;
printf("Enter number of elements in array:");
scanf("%d", &n);
printf("enter %d integers", n);
for(i=0; i<n; i++)
{
   if(j=i+1, j<n; j++)
   {  if(arr[i] < arr[j])
      {  if(temp = arr[i]);
         {
            arr[i] = arr[j];
            arr[j] = temp;
         }
      }
   }
printf("In elements of array is sorted in ascending
          order:");
for(i=0; i<n; i++)
{printf("%d", arr[i]);
}
printf("Enter value to find");
```

```
scanf ("%d", &key);
low = 0
high = n-1
mid = (low + high)/2 ;
while (low < high)
{
  if (arr[mid] > key)
{
  low = mid+1;
}
  else if (arr[mid] = key)
  {
    printf ("%d found at location %d", key, mid+1);
    break;
  }
  else
  high = mid-1
  mid = (low + high)/2
  }
  if (low > high)
  {
  printf ("Not found! %d isn't present in list n", key);
  }
  printf("\n");
  printf("enter two locations to find sum and
        product of the elements");
  scanf ("%d", &one);
  scanf ("%d", &two);
  sum = (arr[one] + arr[two]);
  product = (arr[one] * arr[two]);
  printf ("The Sum of elements = %d", sum);
  printf ("The product of elements = %d", product);
  return 0;
}
```

(3) Insertion sort: Insertion sort algorithm picks elements one by one and places it to the right position where it belongs in the sorted list of elements.

Example for insertion sort:

input - 89 17 8 12 0

step1: 89 17 8 12 0

step2: 17 89 8 12 0

step3: 8 17 89 12 0

step4: 8 12 17 89 0

step5: 0 8 12 17 89

program:

```
#include <stdio.h>
int main()
{
int i, j, count, temp, number [25];
printf ("How many numbers are going to enter? :");
scanf ("%d", &count);
printf ("Enter %d elements :", count);
for (i=0; i<count; i++)
{
scanf ("%d", &number [i]);
for (i=1; i<count; i++){
scanf ("%d", &number [i]);
for (i=1; i<count; i++){
temp = number [i];
j = i-1;
while (temp< number [j]&& (j>=0) ){
number [j+1] = number [j];
j = j-1;
}
number [j+1] = temp;
}
```

```
printf ("order of sorted elements :");
for(i=0; i<count; i++)
    printf ("%d", number[i]);
return 0;
}
```

Selection sort : The smallest element is exchanged with first element of the unsorted list of elements. Then, second smallest element is exchanged with the second element of the unsorted list of elements and so on until all the elements are sorted.

example,

Input - 22 0 -90 89 17

i) -90 0 22 89 17

ii) -90 0 22 89 17

iii) -90 0 17 89 22

iv) -90 0 17 22 89

program :

```
#include <stdio.h>
int main(){
int i, j, count, temp, number[25];
printf ("No. of elements :");
for(i=0; i<count; i++)
   scanf ("%d", &number[i]);
for(i=0; i<count; i++){
      for(j=j+1; j<count; j++){
         if (number[i] > number[j]){
            temp = number[i];
            number[i] = number[j];
            number[j] = temp;
         }
      }
}
```

```
printf ("Sorted elements :");
for (i=0; i< count ; i++)
        printf ("%d", number [i]);
return 0;
}
```

④
```
#include <stdio.h>
#include <conio.h>
int main()
{
int arr [sa], i, j, n, temp, sum = 0, product = 1 ;
printf ("enter total elements to be stored :");
scanf ("%d", &n);
printf ("enter %d elements :", n) ;
for (i=0; i<n ; i++)
scanf ("%d", &arr[i]);
printf ("\n sorting array using bubble sort");
for (i=0; i<(n-1); i++)
{
for (j=0; j<(n-i-1); j++)
  {
    if (arr[j] > arr[j+1]
    {
      temp = arr[j] ;
      arr [i] = arr [j+1];
      arr [j+1] = temp ;
    }
  }
}
printf ("All array elements sorted successfully");
printf ("All elements in ascending order :");
```

```
for (i = 0; i<n; i++)
{
  printf ("%d", arr[i]);
}
printf ("array elements in alternate order");
  for (i = 0; i<=n; i = i+2)
  {
    printf ("%d", arr[i]);
  }
for (i = 1; i<=n; i = i+2)
  {
    sum = sum + arr[i];
  }
printf ("The sum of odd position elements are
                = %d", sum);
for (i = 0; i<=n; i = i+2);
{
  product = arr[i];
}
printf ("The product of even position elements are
                = %d", product);

  getch();
return (0);
}
```

output:

Enter total number of elements to be stored = 4

Enter 4 elements

2

9

1

3

Sorting array using bubble sort

All array elements sorted successfully

Array elements in ascending order.

1
2
3
9
The sum of odd position element is 4
The product of even position element is 18

(2)
```
#include <stdio.h>
#include <conio.h>
#define MAX_SIZE 5
void merge_sort [MAX_SIZE];
void meaige_array (int, int, int, int);
int arr_sort [MAX_SIZE];
int main()
{
int i, k, pro = 1;
printf ("Sample merge sort example functions and
                                    array");
printf ("Enter %d elements for sorting MAX_SIZE);
for (i=0; i< MAX_SIZE ; i++)
{
scanf ("%d", &arr_sort[i]);
printf ("In your data");
}
for (i=0; i< MAX_SIZE ; i++)
{
printf(" %d", arr_sort [i]);
}
merge sort (0, MAX_SIZE - 1);
printf ("sorted data:");
for (i= 0; i< MAX_SIZE ; i++)
{
```

```
printf("%d", arr_sort[i]);
}
printf("Find the product of kth element from first,
        last when k ");
scanf("%d", &k);
Pro = arr_sort[k] * arr_sort[MAX_SIZE - k - 1];
printf("product = %d", pro);
getch();
}
void merge_sort(int i, int j)
{
int m;
if(i<j)
{m = (i+j)/2;
merge_sort(i, m);
merge_sort(m+1, j);
merge_array(i, m, m+1, j);
}
}
void merge_array(int a, int b, int c, int d)
{
int t[50];
int i=a, j=c, k=0;
while(i<b && j<=d)
{if(arr_sort[i] < arr_sort[j])
t[k++] = arr_sort[i++];
else
t[k++] = arr_sort[j++];
}
while(i<=b)
```

```
    t [k+1] = arr_sort [j++];
for (i=a; j=a, ic=d; i++; j++)
arr_sort [i] = t [j];
}
```

(5.) # C program to perform Binary Search using Recursion.

```
#include <stdio.h>
void binary_search(int [], int, int);
void binary_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("Enter key to search :");
 scanf("%d", &key);
 binary_search(list, 0, size, key);
}
void binary_search(list[], int size)
{
 int temp, i, j;
 for(i=0; i<size; i++)
 {
    for(j=1; j<size; j++)
    {
       if(list[i] > list[j])
       {
          temp = list[i];
          list[i] = list[j];
          list[j] = temp;
       }
    }
 }
}
```

```
void binary_search (int list[], int lo, int hi, int key)
{
  int mid;
  if (lo > hi)
  {
   printf ("key not found");
   return;
  }
  mid = (lo+hi)/2;
  if (list [mid] == key)
  {
    printf ("key found");
  }
  else if (list [mid] > key)
  {
   binary_search (list lo, mid-1, key);
  }
  else if (list [mid] < key)
   {
   binary_search (list, mid+1, hi, key);
   }
}
```