Assignment-6

Sanjay V
AP19110010492
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 the number of elements in array");
scanf("%d", &n);
printf("Enter %d integers: ", n);
 for (i = 0; i <n ;i++);
scanf("%d", & arr[i])
 for(i =0;i<n ; i ++)
 {
  if (j =i+1 ;j<n ;j++)
    {
     if(arr[i] < arr[j])
     {
       if (temp = arr[j]);
          {
           arr[i] = arr[j];
          arr[j] =temp;
     }
  }
}
```

```
printf("In elements of array is sorted in decending order:\n");
for(i=0;i<n;i++)
 {
  printf("%d", arr[i]);
  }
printf("Enter the 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 the list.
                                    \n", key);
}
printf("\n");
printf("Enter two locations to find sum and product
                                    of the element")

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;
}
```

Output:

```
Enter number of elements in array: 5
Enter 5 integers
 9
 7
 5
 4
 2
Elements of the array is sorted in decending
                                    order.
```

97542 Enter value to find 5

5 found at location 3

Enter two locations to find sum and product of the elements

2

4

The sum of elements = 67

The product of elements = 10

(2)
```
#include<stdio.h>
#include<conio.h>
#define MAX_SIZE 5
void merge_sort[MAX_SIZE];
Void merge_array(int, int, int, int);
int ar-sort[MAX_SIZE];
int main()
{
int i, k, Pro = 1;
Printf("Sample merge sort example funcing
                and array \n");
printf("\n Enter %d Elements for sorting\n",
                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("\t %d", arr_sort[i]);
}
merge_sort(0, MAX_SIZE-1);
printf("\n sorted data:");
for(i=0;i<MAX_SIZE;i++)
{
printf("\t %d", arr_sort[i])
}
printf("Find the product of the kth element from
first and last where k\n");
scanf("%d", &k);
pro = arr_sort[k] * arr_sort[MAX_SIZE-k-1];
printf("Procedure = %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);

//merging two arrays
  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++];
 }
//collect remaining elements
  while(i<=b)
     t[k++]=arr_sort[j++];
```

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

Output:-

Sample Merge sort example - functions and arrays

Enter 5 elements for sorting

9
7
4
6
2

your data: 9 7 4 6 2

sorted data: 2 4 6 7 9

Find the kth elements from first and last

where k = 2

product = 36

(3) Insertion Sort: Insertion sort works by inserting the set of values in the existing sorted file. It constructs the sorted array by inserting a single element at a time. This process continues until whole array is sorted in the same order. The primary concept behind insertion sort is to

insert each item into its appropriate place in the final list. The insertion sort method saves an effective amount of memory.

Ex: arr[] = 46 22 11 20 9

// Find the minimum element in arr and place at begining

9 46 22 11 20

// Find the 2nd minimum element in the arr and insert in between the begining

9 11 46 22 20

// Repeat above steps and at last

9 11 20 22 46.

~~Sorted~~ Selection Sort: Selection Sort performs sorting by searching for the minimum value number and placing it into the first or last position according to the order. The process of searching the minimum key and placing it in the proper position is continued until all elements are

placed at right position.

Ex: 13 12 4 6 7

let us loop for i=1 to 4

i=1, since 12 is smaller than 13, move 13

and insert 12 before 13.

do same for i=2, i=3, i=4

∴ 4 6 7 12 13

(4)
```
#include <stdio.h>
#include <conio.h>
int main()
{
  int arr[50], i, j, n, temp, sum=0, product=1;
printf("Enter total number of elements to store:")
scanf("%d", &n);
printf("Enter %d elements:", n);
for(i=0; i<n; i++)
scanf("%d", &arr[i]);
printf("%d\n Sorting array using bubble sort");
for(i=0; i<(n-1); i++);
{
```

```
for (j=0; j<(n-i-1); j++)
  {
    temp = arr[j];
    arr[j] = arr[j+1];
    arr[j+1] = temp;
  }
 }
}
printf("All array elements sorted successfully \n");
printf("Array elements in assending order: \n\n");
 for (i=0; i<n; i++)
    {
      printf("%d \n", arr[i]);
    }
printf("array elements in altemate order\n");
   for (i=0; i<=n; i=i+2)
    {
      printf("%d \n", arr[i]);
    }
 for (i=1; i<=n; i=i+2)
   {
     sum = sum + arr[i];
   }
```

```
printf("The sum of odd position elements are =
                %d \n", sum);
for(i=0; i<=n; i+=i+2)
{
 product *= arr[i];
}
printf("The product of even position element
                are = %d \n", product);

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

Output:

Enter total number of elements to store = 5

Enter 5 elements

8 6 4 3 2

Sorted array using bubble sort

All array elements sorted successfully

Array elements in assending order

2 3 4 6 8

array in alternate order

2 4 8

The sum of odd position is 7
The product of even position are 6,4

(5)

```
#include <stdio.h>
#include
void binary search(int arr[], int num, int first,
                   int last)
{
  int mid;
  if (first > last)
    {
     printf("Number is not found");
    }
  else
    {
     mid = (first + last)/2;
    }
  if (arr[mid] == num)
    {
     printf("Element is found at index %d",
                                    mid);
     exit(0);
    }
else if (arr[mid] > num)
  {
   primary search (arr, num, first mid-1);
```

```
    }
  else
    {
    Binary search (arr, num, mid+1, bot);
    }
  }
}
void main()
  {
int arr[100], beg, mid, end, i, n, num;
  printf("Enter the size of an array");
  scanf("%d", &n);
  printf("Enter the value in sorted sequence\n");
  for(i=0; i<n; i++)
  {
  scanf("%d", &arr[i]);
  }
  beg=0
  end=n-1;
  printf("Enter a value to search: ");
  scanf("%d", num)
  Binary search(arr, num, beg, end);
}
```

Output :-

Enter the size of array 5

Enter the values in sorted sequence

4
5
6
7
8

Enter a value to search : 5

Element is found at index : 1