Assignment-6

Searching and Sorting

AHBADI. KOUSHIK
AP19110010512

1) Take elements from user and sort them in descending order

Code:

```
#include <stdio.h>
void descend ( )
void binary search( )
void add_and_mul ( )
main ( )
{
  int choice;
   while(1)
   {
    printf ( "1. Descending order\n ");
    printf ( " 2. searching element in array);
    printf ( " 3. add and multiply \n ");
    printf ( " 4. quit \n ");
     printf (" Enter your choice ");
     scanf (" %d", &choice);
     switch (choice)
     {
      Case 1:
        deseend( )
        break;
     Case 2:
       binary search ( )
```

```
        break;
    Case 3:
        add_and_null()
        break;
    Case 4:
        exit(1);
        default:
        printf("wrong choice in");
    }
}
}
void descend()
{
    int userarray[100], value, i, j, swp;
    printf("Input value");
    scanf("%d", &value);
    for(i=0; i<value; i++)
    {
        printf("value %d:", i+1);
        scanf("%d", &userarray[i]);
    }
```

```
for (  j=0; j < (value-1); j++)
  {  for ( i=0; i<(value-1) ; i++ )
   {
    if (userarray[i+1] < userarray[i])
    {
      swap = userarray[i];
      userarray[i] = userarray[i+1];
      userarray[i+1] = swap;
    }
   }
  }
printf("Descending order: \n");
 for (i=value; i>0; i--)
 {  printf("%d", userarray[i-1])
 }
 return 0;
}
void binary search()
{  int c, fir, las, mid, n, search, array[100];
 printf("Enter no.of element\n");
   scanf("%d", &n);
```

```
printf ("Enter %d integers \n", n);
for (c=0; c<n; c++)
{
scanf ("%d", &array[c]);
}
printf ("Enter value to be find \n");
scanf ("%d", &search);
fir=0
las=n-1;
mid= (fir+las)/2;
while (fir<las)
{
if (array[mid] < search)
{
fir=mid+1;
}
else if (array[mid]==search)
{
printf ("%d found at location %d \n", search,
                                        mid+1);
break;
}
else
{ las=mid-1;
mid = (fir+las)/2;
}
```

```
if (fir > las)
{
printf ("Not found ! %d isn't present in
                                list , search);
  }
}
}
}
void add_and_mul()
{
int x, y, add, multi
printf ("Enter x location: ");
scanf ("%d", &x);
printf ("enter y location:");
scanf ("%d", &y);
add = array[x] + array[b];
multi = array[x] * array[y];
printf ("addition = %d", add);
printf ("multiplication = %d", multi);
}
```

output:

1. Descending order
2. searching element in arrray
3. add and multiply
4. Quit

Enter your choice: 1

Input value: 5

value-1 : 1

value-2 : 4

value 3: 7

value 4: 8

value 5 : 3

Descending order : 8 7 4 3 1

Enter your choice : 2

Enter no. of elements: 5

Enter 5 integers.

8
7
4
3
1

Enter value to find

4

4 found at location 3.

Enter your choice : 3

```
Enter x location : 2
Enter y location : 3
addition = 11
multiplication = 28
Enter your choice : 4
EXIT
```

2) Merge sort:

```
#include<stdio.h>
#define SIZE 100

int inputarray[SIZE];
int secondarray[SIZE];

void merge(int least, int middle, int high)
{
 int i, j, k;
for(i= least, j= middle+1; k= least;
    i<middle && j<=high; k++);
  {
   if(inputarray[i] < inputarray[j])
    {
    secondarray[k] = inputarray[i++];
    }
   else
    {
    secondarray[k]= inputarray[j++];
    }
```

```
}
while (i<=middle)
{
secondarray[k++]= inputarray[i++];
}
while (j<=high)
{
secondarray[k++] = inputarray[j++];
}
for (int i=0; i<high+1; ++i)
{
inputarray[i]= secondarray[i];
}
}
void sort (int least, int high)
{
if (least<high)
{
int middle = (least+high)/2;
Sort (least, middle);
Sort (middle+1, high);
merge (least, middle, high);
}
```

```
else
  return;

}
int main(void)
{
  int n;
  printf("Enter no. of elements");
  scanf("%d", &n);
  printf("Enter %d elements", n);
  for(int i=0, i<n; i++)
  {
    scanf("%d", &inputarray[i]);
  }
printf("\n Array after sorting is :");
  sort(0, n-1);
  for(int i=0; i<n; i++)
  {
    printf("%d", inputarray[i]);
  }
  int k; multi=1;
printf("enter K value \n");
  scanf("%d", &k);
```

```
for (i=0; i<k; i++)
    {
      multi = multi*i
    }
  printf(" product of kth elements is %d", multi);
}
```

Output:

Enter no. of element 3.

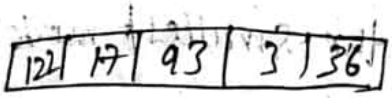
6
9
8
2

Array after sorting is 2 3 6 8 9

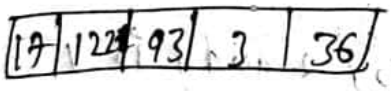
enter k value
3
product of kth elements is 36.

③ Insertion Sort:

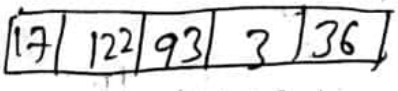
① If the element in first place is already sorted.
② Move to next element.
③ Compare the current element with all elements in sorted array
④ If the element in sorted array is smaller than current element, iterate to the next element otherwise shift all the greater elements in array by one position towards right.
⑤ Insert the value at the ~~correct~~ current position.
⑥ Repeat until the complete list is sorted.

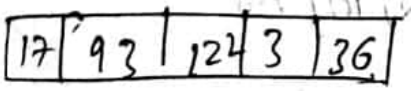


→ 122, 17, 93, 3, 36



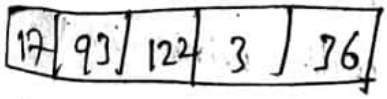
for i = 1 (2nd element)
since 17 is smaller than 122
move 122 and insert 17 before 122.



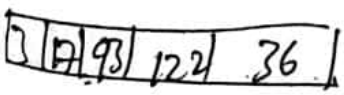


→ 17, 122, 93, 3, 36

For i = 2
Since 93 is smaller than 122, move 122 and insert 93 before 122



→



For i = 3
3 will move to the beginning

and all other elements from 17 to 122 will move one position ahead of their present position

→ 3, 17, 73, 122, 36.

for i = 4 36 will move to position after 17, and element from 93 to 122 will move one position towards Right

→ 3, 17, 36, 93, 122

Selection Sort:

Consider array [10, 5, 2, 1]

The first element is 10.

The next part we must find the smallest number from remaining array.

The smallest number from 5, 2 and 1 is 1

So, we replace 10 by 1

The New array is [1, 5, 2, 10] Again, this process is repeated

Finally we get the sorted array as [1, 2, 5, 10]

→ Set minimum to first location

→ search minimum element in array

→ Swap the first location with minimum value in array

→ Assign the second element as min
→ Repeat the process until we get sorted array

(8) Bubble Sort:

```
#include<stdio.h>
int Bubblesort (int size, int *array)
{
  int i, j, temp;
  for (i=size-2; i>=0; i--)
  {
    for(j=0 ; j<=i ; j++)
    {
      if (array(j) > array[j+1])
      {
        temp = array [j];
        array [j] = array [j+1];
        array [j+1] = temp;
      }
    }
  }
  return;
}
```

```
int main(void)
{
 int size, array[20], sum=0, mul=1,i;
 printf("Enter total no. of elements\n");
  scanf("%d", &size);
 printf("Enter the %d elements: ", size);
 for(i=0; i<size; i++)
     {
      scanf("%d", &array[i]);
     }
 BubbleSort( size, array);
  printf("After sorting ");
 for(i=0; i<size ; i++)
  {
    printf("%d", array[i]);
   }
 printf("\n");
  printf("alternate elements after sorting\n");
  for(i=0 ; i<size ; i++)
    {
     printf("%d", array[i++]);
    }
   printf("\n");
```

```
printf ("sum of elements in odd positions and multi
    of elemnt in even position");
for (i=0; i<size; i++)
{
  if (i%2==0)
  {
    mul=mul*array[i]
  }
else
  {
  sum=sum+array[i]
  }
}
}
printf ("sum of element in odd position is %d", sum);
printf ("mul of element in even position is %d", mul);
printf ("enter m value");
scanf ("%d", &m);
for (i=0; ; i<size; i++)
{ if (array[i]%m==0)
    {
      printf ("%d", array[i]);
    }
}
printf ("\n");
```

```
return 0;
}
```

output:

```
Enter total no. of elements: 5
Enter the 5 elements = 3
   5
   7
   1
   9
After Sorting 1, 3, 5, 7, 9
Alternate elements after sorting
   1   5   9
Sum of elements in odd positions and multiplication
of elements in even positions
Sum of elements in odd positions is 15
mul of elements in even positions is 21
   enter m value is: 3
      3
      9
```

6) Recursive Binary Search:

```
#include <stdio.h>
# define size 10
int binary search ( int a[ ], int key, int low, int
                                              high)
{
  int mid;
   mid = (low+high)/2;

   if (key == a[mid])
        return mid;
    else if (key < a[mid])
                binary search (a, key, low, mid-1);
     else
       binary search (a, key, mid+1, high);
    }
   void main( )
   {
   int a[size], key, n, i, flag = 0;
printf ("\n, How many elements are there in
        array ");
 scanf ("%d", &n);
printf (" enter the elements ");
  for (i=0; i<n ; i++)
     {
       scanf ("%d", &a[i]);
     }
```

```
printf ("Enter the element to be searched");
scanf ("%d", &key);
flag = binary search (a, key, 0, n-1);
printf ("Element is found at %d position", flag+1);
}
```

output :

How many elements are there : 5

Enter elements: 1
2
3
6
5

Enter the element to be searched : 6

Element is found at 4th position